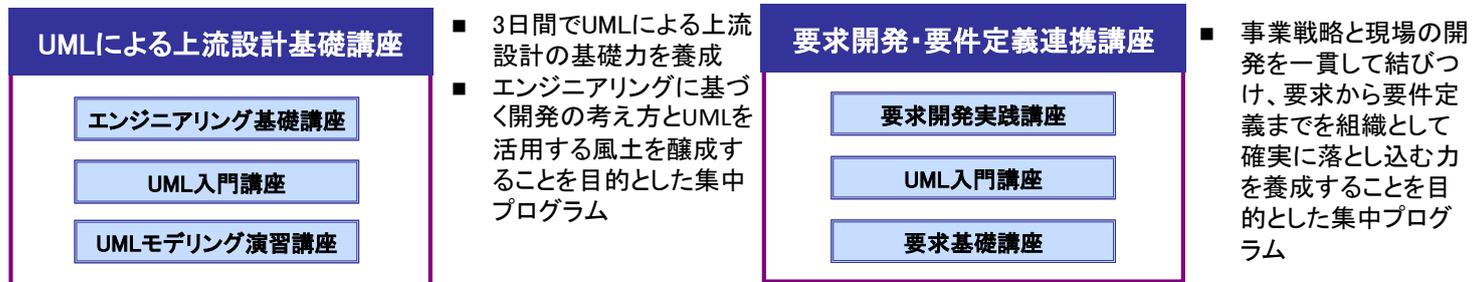
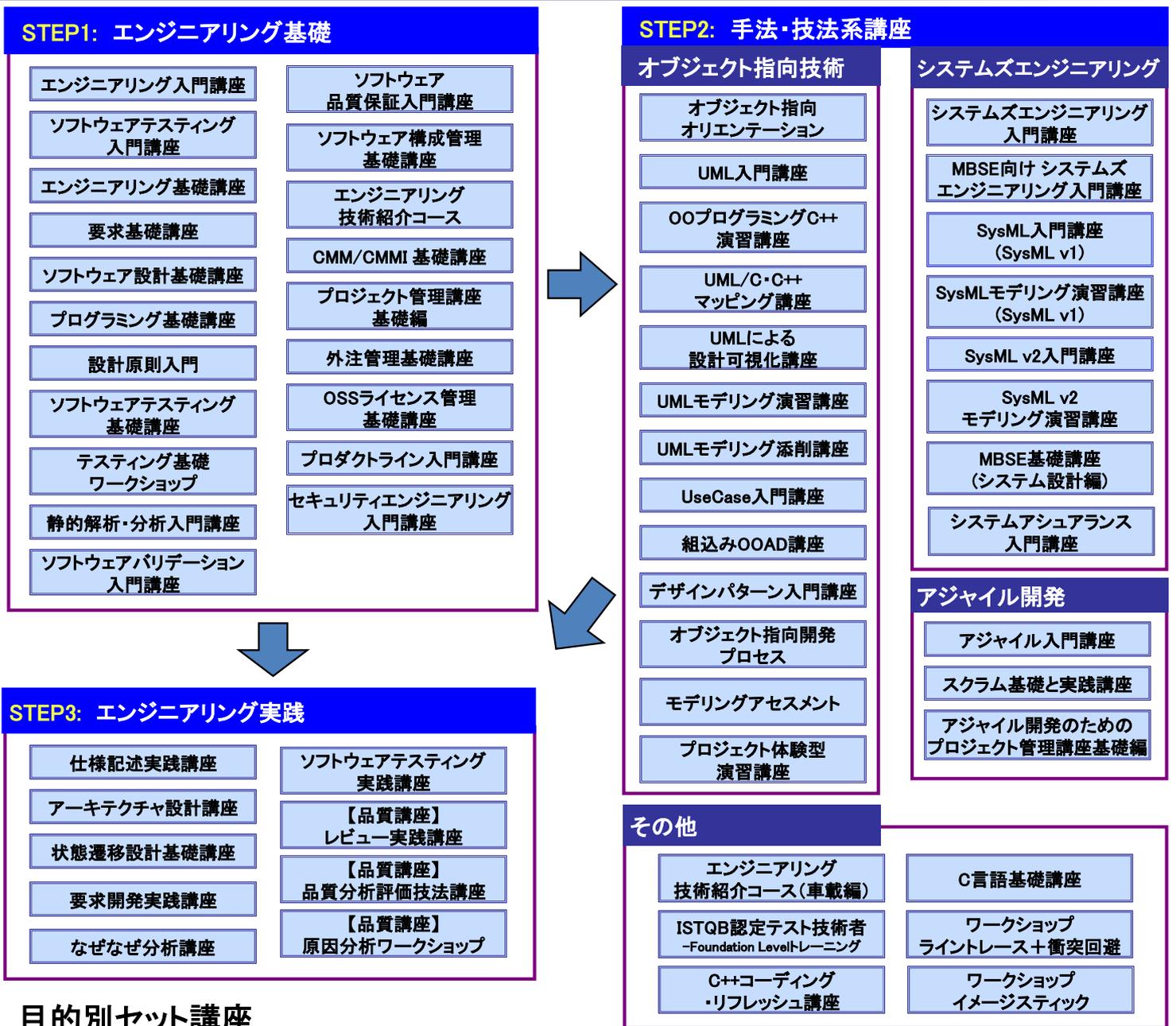


エンジニアリング教育

組込みエンジニア向け 標準教育コースの構成



管理者・発注者向け 教育コース



エンジニアリング教育の特長

- ソフトウェアエンジニアリング全体を網羅したコースウェアの提供
- 定評あるオブジェクト指向技術教育コースに加え、エンジニアリングの基本(原理原則)を教えるコースウェアの提供
- オブジェクト指向技術の”エンジニアリング”レベルでとりあげた実践的なコースウェアの提供
- 教科書的でなく、開発現場の実態に即したコースウェア
- 現場で直面する課題を題材にした演習中心のコースウェア
- コースウェア作成から講師まで、実践経験のあるコンサルタントが担当
- 経験豊富なコンサルタントが、受講者層や目的に応じて柔軟に講義を実施

- 講座開発
 - ご要望に応じて御社向けに特化した講座を作成します
 - 御社内で講座が実施できるよう講師育成をご支援します
- 教育体系構築支援
 - 御社のエンジニアに必要なスキル要件を洗い出し、それに対して必要な知識項目を割り付け、それを踏まえて教育体系を構築します
- スキル診断
 - 御社のエンジニア一人一人のスキル診断をソフトウェアエンジニアリングの各知識領域毎に実施します
- 新人教育
 - マイコン基礎、C/C++言語、組み込みOSからUML、オブジェクト指向開発まで、ご要望に応じた新人教育をご提案・実施します
 - 弊社の新人教育は、以下の方針、考え方で実施します
 - 『開発基礎力』の獲得、『考える力』の醸成、実践的な仮想体験、工学的職人気質
- 転換者教育
 - メカ・エレキ設計者からソフトウェア設計者への転換教育をご要望に応じて、ご提案・実施します
 - メカ・エレキ・ソフト設計者からシステム設計者への転換教育をご要望に応じて、ご提案・実施します

教育内容等ご要望がございましたらお問合せください

※ 問題・課題に応じた最適な講座をご提案いたします

※ ご要望に応じて講座日数、テキストや演習問題をカスタマイズすることができます

カスタマイズ範囲:

- ・ テキストの追加・変更、演習問題の追加・変更
- ・ 特定ツールへの対応(例: 演習をモデリングツールを使う形に変更)
- ・ 開催時間の短縮・延長、講義後のディスカッションの追加

※ 複数日開催の講座を1週間に1日または1ヶ月に1日ずつ開催することも可能です

※ 開催形式をお選びいただけます。

- ・ お客様の会場に講師が出向くオンサイト開催、オンライン会議システムを使用したオンライン開催
- ・ 講座開催費は、オンサイト開催・オンライン開催共に同額となります。但し、オンサイト開催の場合、場所によっては諸経費(宿泊費、交通費)がかかります。

※ 受講料は、受講者20名まで同料金。受講者が20名を超える場合は費用が変わります。

株式会社 豆蔵 営業本部
〒163-0434 東京都新宿区西新宿2-1-1 新宿三井ビル34F
TEL: 03-5339-2136 月～金 9:30～18:00(土・日・祝日除く)
E-mail: sales@mamezou.com
URL : <https://www.mamezou.com/>



STEP1 : エンジニアリング基礎

エンジニアリング基礎の講座の基本構成

■ エンジニアリングの基礎(プロセス構成、各プロセスで実施事項、要点)を習得する

#	講座名	内容	所要 日数	ページ
1	エンジニアリング入門講座	初めてソフトウェア開発を実施するに当たり、開発の進め方、各活動の必要性、および開発チームの中でどのように振舞わなければならないのかを学ぶ。	1日間	8
2	ワークショップ	講座受講後、自社/自部門の開発と標準的な開発の差異を認識し、どうすれば良い開発に変えることができるかをグループ討議することで、エンジニアリングの重要性を認識するワークショップ。	—	
3	ソフトウェアテスト入門講座	ソフトウェアテストの必要性や、テストを実施するに当たり必要となる基本的な知識、進め方、およびテスト設計スキルを学ぶ。	2日間	8
4	エンジニアリング基礎講座	システムエンジニアリング、ソフトウェアエンジニアリングの中で行われる活動の目的/役割/実施内容を理解し、それらを現実の活動に落とし込むにはどうすればよいかを学ぶ、基礎的なコース。	1日間	9
5	要求基礎講座	要求の特性や性質、要求領域での活動の体系といった要求の基礎、要求開発の位置づけなどを学ぶ。演習では、具体的な問題を利用して、要求のレビューや分析を体験します。	1日間	9
6	ソフトウェア設計基礎講座	設計の基本的な考え方やポイント、設計原則といった設計の基礎を学びます。構造化設計やオブジェクト指向設計などの手法に依存せず、良い設計の本質を理解し、実務に活かせる力を養うことができます。	1日間	10
7	プログラミング基礎講座	特定のプログラミング言語によらない、ソフトウェア構築(ソフトウェア・コンストラクション)の基本的な考え方、要点、進め方、原則/ガイドラインといった基礎の習得にフォーカスする実装担当者向けコース。	1日間	10
8	設計原則入門講座	オブジェクト指向設計の11原則を体系的に学び、拡張性・再利用性を高める実践力を養う2日間講座。	1日間 2日間	11
—	ソフトウェアテスト基礎講座	ソフトウェアテストの進め方とテスト設計技法に対する基本的な知識を習得する、実践的な教育コース。	—	11
9	第Ⅰ部：プロセス編	ソフトウェアテストの全体構成と、各活動の目的と進め方を具体例と演習を通して学ぶ。	1日間	
10	第Ⅱ部：技法編	基本的なテスト技法を体系的に紹介し、演習を通してソフトウェアテスト設計スキルを習得する。	1日間	
—	テスト基礎ワークショップ	ソフトウェアテストの進め方とテスト設計技法に対する基本的な知識を習得する、実践的な教育コース。	—	12
11	計画編	ソフトウェアテストプロセスでも重要な、テスト計画の考え方を、具体的な題材を用いた演習で体得する。ソフトウェアテスト基礎第Ⅰ部(全体テスト計画)に対応。	1日間	
12	設計編	効果的・効率的なテストに欠かせないテスト分析・テスト設計の進め方を、具体的な題材を用いた演習を通じて学ぶ。ソフトウェアテスト基礎第Ⅰ部(テスト分析・設計)、第Ⅱ部(テストケース設計技法)に対応。	1日間	
13	静的解析・分析入門講座	静的分析の必要性、静的解析・分析でわかること、静的分析の方法および不具合の修正方法を学ぶ。静的分析結果とプログラムコードの関係、プログラミングの原理・原則に関する解説も含む。	1日間	12
14	ソフトウェアバリデーション入門講座	ソフトウェア品質保証活動における「妥当性確認」をテーマとした講座。妥当性確認は、一般的な品質保証活動だけでなく、安全規格においても、その実行が求められているが、具体的にどのような活動を行えばよいかの分かりにくい活動である。本講座は安全規格対応も含めた形で、妥当性確認の目的やその活動について具体的に説明していく。	1日間	—
15	ソフトウェア品質保証入門講座	ソフトウェア品質保証を、ソフトウェア開発の一部としての活動という観点から整理し、基礎概念や代表的な活動・技法、および品質保証計画の重要性を学ぶ。	1日間	13
16	ソフトウェア構成管理基礎講座	ソフトウェア構成管理の基本概念、ソフトウェア構成管理の各活動での実施事項・要点、構成管理環境の構築等を学ぶ。	1日間	13
17	エンジニアリング技術紹介コース	世の中にある代表的なエンジニアリング技術について、その技術の狙い/概要/効果を紹介するコース。	1日間	14

#	講座名	内容	所要 日数	ページ
—	CMM/CMMI講座	具体的な活動イメージを示しながら、CMM/CMMIのポイントを解説していく、実践的なCMM/CMMI教育コース。	—	—
18	CMM/CMMI 講座_レベル2 _SEPG向け	具体的な活動イメージを示しながら、CMMの基本概念、全体構造、並びにCMMレベル2の各プロセス領域の概要を学んでいくSEPG向けコース。	1日間	15
19	CMMI 講座_レベル3 _SEPG向け	具体的な活動イメージを示しながら、CMMIのレベル3の全体像、各プロセス領域のゴールの内容・レベル、各プラクティスの概要・ポイントを学んでいくSEPG向けコース。	1日間	—
20	CMMI 講座_レベル2_概要	具体的な活動イメージを示しながら、CMM/CMMIの基本概念、全体構造、並びにCMMIレベル2の各プロセス領域の概要を学んでいく開発者向けコース。	1日間	—
21	CMMI 講座_レベル3_概要	具体的な活動イメージを示しながら、CMMIレベル3の各プロセス領域の概要を学んでいく開発者向けコース。	1日間	—
22	プロジェクト管理講座 基礎編	プロジェクトを円滑に推進していくのに必要な管理の基本（要件管理、計画、進捗管理）・実践上の要点を習得する。	1日間	15
23	外注管理基礎講座	外注を使用している発注側の方を対象に、外注管理を円滑に推進していくのに必要な管理の基本・実践上の要点を理解するコース。	1日間	16
24	OSSライセンス管理基礎講座	OSSライセンス、プロセス構築方法と最近の動向を基礎から解説する、ソフトウェア開発を含む製品開発を管理されるマネージャーと、オープンソースソフト(OSS)管理プロセスを構築される方向けのコース。	0.5日 ~1日間	16
25	ソフトウェアプロダクトライン入門講座	基本的活動とプラクティスエリアのエッセンスを講義と確認テストで学ぶ。	1日間	17
26	セキュリティエンジニアリング 入門講座	「セキュリティエンジニアリング」として、製品開発/システム開発の中でどのようにセキュリティの観点に対応していくのかについて解説。セキュリティ対策における重要なアプローチである、セキュリティ・バイ・デザインに基づき、開発ライフサイクル(要求獲得-仕様定義-アーキテクチャ設計-詳細設計-構築-テスト)内でセキュリティリスクをどのように扱っていくのかを学ぶ。	1日間	17

STEP2 : 手法・技法系講座

オブジェクト指向技術の講座の基本構成

■ 技術(= オブジェクト指向技術)それ自身を習得する

#	講座名	内容	所要 日数	ページ
27	オブジェクト指向オリエンテーション	オブジェクト指向の基本的な考え方や基礎概念、並びにオブジェクト指向の有用性を習得する。	1日間	18
28	UML入門講座	UML(Unified Modeling Language)2.0について、実践で必要となる基本的な各ダイアグラムの表記法を演習もまじえ、学習する。	1日間	18
29	オブジェクト指向プログラミング C++演習講座	オブジェクト指向プログラミング(C++言語)の基礎(考え方・進め方・ポイント)を学ぶ。	4日間	19
30	UML/C・C++マッピング講座	UML(クラス図、ステートマシン図、シーケンス図)とプログラミング言語(C言語またはC++)の対応関係を演習を通して学ぶ。プログラミング言語は、いずれかを選択して開催。	1~2日間	19
31	UMLによる設計可視化講座	UMLとソースコードの対応関係を学び、設計を可視化する方法を習得します。フォワード/リバースエンジニアリングを通じ、全体像の理解や影響範囲の把握、設計改善を実現する講座です。	1~2日間	20
32	UMLモデリング演習講座	モデリングの考え方、並びにオブジェクト指向のモデリングの基本を演習を通じ学ぶ。	1~2日間	20
33	UMLモデリング添削講座	UMLモデリングスキルの向上を目的とした、通信添削講座。	1.5~ 7.5ヶ月	21
34	UseCase入門講座	UseCaseを用いた要求定義方法、UseCase分析について学ぶ。	1日間	21
35	オブジェクト指向分析設計講座	組込みソフトウェア開発に則した演習を通じ、オブジェクト指向の分析、設計の基本を学ぶ。	3日間	22

#	講座名	内容	所要 日数	ページ
—	デザインパターン入門講座	デザインパターンの目的・概要、並びにパターンの適用方法の基本を演習を通して学ぶ。		22
36	概要編	デザインパターンの概要を説明し、主に開発で用いられるデザインパターンの解説・利用法を学ぶ。	1日間	
37	適用編	題材をベースに、課題、課題と解決するデザインパターン、適用時のポイントを学ぶ。	1日間	
38	コンパクト版	概要編と適用編をコンパクトにまとめ、1日間で効率的に学習する講座	1日間	
39	オブジェクト指向開発プロセス講座	オブジェクト指向の代表的な開発プロセスであるUP (UnifiedProcess)、XP (eXtreamPrograming)、並びに反復型開発プロセスの要点について学ぶ。	1日間	23
40	モデリングアセスメント講座	受講者のUMLモデリングスキルを豆蔵独自の「モデリングに必要な力」から診断。	診断結果 が出るまで 1週間	23
41	プロジェクト体験型演習講座	Arduinoを使って、組み込みソフトウェアの基本+SWエンジニアリングの基本を学ぶ。ブレッドボード、センサ、LED等も用意しており、50名程度まで対応可能。	3日間	24

システムズエンジニアリング技術の講座の基本構成

■ 技術(= システムズエンジニアリング技術)それ自身を習得する

#	講座名	内容	所要 日数	ページ
42	システムズエンジニアリング入門講座	システムズエンジニアリングについて、その意義と 活動内容について学びます。またモデルを活用したシステムズエンジニアリングであるMBSE(Model Based Systems Engineering)とはどのようなものかについても学習する。	1日間	24
43	SysML入門講座 (SysML v1)	システムエンジニアリング向けの標準モデリング言語として注目されているSysML(Systems Modeling Language)の概要を理解し、実践で必要となる基本的な表記法について、演習をまじえて学習する。	1日間	25
44	SysMLモデリング演習講座 (SysML v1)	システム工学の基本的な概念とSysML表記法を習得した人向けの「次のステップ」の講座。「知識」として得た情報を「モデリング・スキル」として定着/使いこなせるようになるために演習/レビューを繰り返す。	1日間	25
45	SysML v2入門講座	システムエンジニアリング向けの標準モデリング言語として注目されているSysML v2 (Systems Modeling Language v2) の概要を理解し、実践で必要となる基本的な表記法を演習を通じて学習する。UMLやSysML v1の未経験者も安心して受講可能。	1日間	26
46	SysML v2モデリング演習講座	システム工学の基本的な概念とSysML v2表記法を習得した人向けの「次のステップ」の講座。「知識」として得た情報を「モデリング・スキル」として定着/使いこなせるようになるために演習/レビューを繰り返す。	1日間	26
47	MBSE基礎講座(システム設計編)	システム設計における、設計課題の抽出、システム構造の記述、システムの振舞いの記述、トレードオフ分析といった活動の中でSysMLをどのように活用するのかを具体的にモデルを作成しながら学習する。	1日間	27
48	システムアシュアランス入門講座	システムアシュアランスの技術を取り上げ、その考え方や使い方について学習する。安全規格対応のみならず、自身の考え方を他者へ正確に説明したいという方に最適の講座。	1日間	27

アジャイル開発

■ アジャイル開発の入門から実践までの一連を習得する

#	講座名	内容	所要 日数	ページ
50	アジャイル入門講座	アジャイル開発とは何か、どのような意義があり、どのような場合にアジャイル開発を採用する必要があるのかを学ぶ。表面的なプロセスの説明では分かりにくい「なぜそうするのか」という根底の部分に迫り、グループディスカッションと演習で理解を深める。	1日間	28
51	スクラム基礎と実践講座	アジャイル開発手法の中でも主流のスクラムをとりあげ、アジャイルの生まれた背景や基本知識、開発の進め方をわかりやすく解説。実践では、演習を通じてスクラム開発の全体像を学習する。	1日間	28
52	アジャイル開発のためのプロジェクト管理講座 基礎編	CMMやPMBOKの要件管理・計画・進捗管理をアジャイルのプラクティスに対応付け、自チームの活動を見直すことで、スクラム実践の計画性と安定性を高める講座です。	1日間	29

STEP3 : エンジニアリング実践

エンジニアリング実践の講座の基本構成

- 基礎の上でのOO技術の有効な使用、あるいはOOを技術を使ってエンジニアリング基礎の実現を習得する

#	講座名	内容	所要 日数	ページ
53	仕様記述実践講座	要件・仕様記述における日本語の使い方、仕様記述の基本と具体的な方法、UML (Unified Modeling Language) のステートマシン図による記述方法について学ぶ。	1日間	29
54	アーキテクチャ設計講座	オブジェクト指向開発におけるアーキテクチャ設計の実際、UMLの活用方法を学ぶ。	1日間	30
55	状態遷移設計基礎講座	演習を通じて状態遷移設計を多角的に習得し、組込み開発に必須の振る舞い設計力を強化。	2日間	30
56	要求開発実践講座	ビジネス課題を起点に要求を定義し、価値ある製品やサービス創出につなげる実践的な3日間講座。	3日間	31
57	ソフトウェアテスト実践講座	オブジェクト指向、UMLを前提にソフトウェアテストの実践方法を演習を通じて学ぶ。(※オブジェクト指向/UML前提)	2日間	31
—	ソフトウェア品質講座	ソフトウェア品質向上/確保に必要な知識・手法を演習を通して習得する。		—
58	レビュー実践講座	レビューの勘所/"レビューの質を向上"させるポイント(欠陥除去の視点・原則・経験則)を演習を交えながら習得する。	1日間	32
59	品質分析評価技法講座	主な品質分析評価技法の基礎ならびにその適用を学ぶ。	1日間	32
60	原因分析ワークショップ	ソフトウェア不具合の原因分析と再発防止における考え方、プロセス、注意点等を、具体的な題材を用いた演習を交えながら習得する。	1日間	33
61	なぜなぜ分析講座	問題をロジカルに分析するツールとして話題の「なぜなぜ分析」を使って製造現場で起こっているヒューマンエラーを分析する際の重要ポイントについて解説します。また、グループワークを行い、分析方法について理解を深めます。	1日間	33

その他

#	講座名	内容	所要 日数	ページ
62	JSTQB認定テスト技術者 Foundation Levelトレーニング	ソフトウェアテストの基本的な手法や技法について、国際的に普及しているISTQBのシラバスに基づいて、効率的に学ぶ。	3日間	34
63	C++コーディング・リフレッシュ講座	C++11以降のModern C++を体系的に学び、効率的で安全なコードを実現する実践的な講座。	3日間	34
64	エンジニアリング技術紹介コース (車載編)	車載システム開発に必要な代表技術をまとめて学び、動向と活用法を効率的に理解できる講座。	1日間	14
65	C言語基礎研修	組込みソフトウェア開発のコーディングの土台となるプログラミング言語「C言語」を基礎から学ぶ。	10日間	35
66	ワークショップ(ライトレース+衝突回避)	STEP1、STEP2で学んだ手法を生かし、仮プロジェクトを実際に運用しながら組込み開発を一通り体験する。	5日間	35
67	ワークショップ(イメージスティック)	弊社オリジナル製品「イメージスティック」を用いた組込み開発を体験する。	10日間	36

目的別セット講座

#	講座名	内容	所要 日数	ページ
68	UMLによる上流設計基礎講座	3日間でUMLによる上流設計の基礎力を養成する。エンジニアリングに基づく開発の考え方やUMLを活用する風土を醸成することを目的とした集中プログラム。	3日間	36
69	要求開発・要件定義連携講座	事業戦略と現場の開発を一貫して結びつけ、組織としてビジネス価値の実現となる要件定義まで落とし込む力を養成することを目的とした集中プログラム。	2日間 3日間	37

管理者・発注者向け 教育コース

マネージャ向け講座

#	講座名	内容	所要 日数	ページ
70	マネージャ向け エンジニアリング基礎講座	ソフトウェア開発経験のないマネージャや経営層を対象に、開発の難しさやソフトウェアエンジニアリングの概要を解説します。要求・設計・テストの要点を短時間で学び、内製化を進める上での要点を学びます。	0.5日間	9
71	マネージャ向け オブジェクト指向分析設計講座	オブジェクト指向を開発に導入した場合の流れとそこで作られる成果物、成果物のチェック方法やマネジメント活動との関連を管理者の視点から学ぶ。	1日間	37

発注者向け講座

#	講座名	内容	所要 日数	ページ
72	発注者向け 要件定義講座	ソフトウェアが専門ではない事業部門の方を対象に、システム発注時に押さえておくべき要件定義のポイントを解説します。	1日間	38
73	発注者向け 受入れテスト講座	事業部門の発注者がシステム受入れを行う際に注意すべき観点やテストの要点を学ぶ講座です。	1日間	38

エンジニアリング入門講座



本講座は、ソフトウェア開発経験のない新入社員向けに、ソフトウェア開発の進め方と、その中で行われる活動の目的と必要性をわかりやすく解説した講座です。新入社員の中には学校でソフトウェア技術を学んだ人もいますし、趣味でソフトウェア開発を経験した人もいます。しかし、企業でのソフトウェア開発は、学校の実習や趣味とは違い、開発には必ず納期があり、また、大規模、複雑化、開発期間の短期化や不確定で変動する要求の対応などの中で開発を行なわなければならないかもしれません。したがって、経験のある新入社員でも企業の開発現場で良いソフトウェアを開発することは容易なことではありません。

この講座は、個人のプログラミングと企業におけるソフトウェア開発との違いや、ソフトウェア開発の全体像を体系的に理解してもらうためのものです。

本講座を受講することによって、これから新入社員が学ぶ設計手法やプログラミング言語、あるいはテスト手法などの個々の技術、あるいは開発を担当する部分の目的、位置づけが明確になり、新入社員がより良いソフトウェア開発を行うことができるようになります。

またワークショップは、受講者が学んだことを利用した題材で検討し、発表・討議を通してエンジニアリングに関する理解度を深めることができます。

是非、新入社員研修を効果的にするために本講座をご利用ください。

日 数

1日

時 間

7時間(9:00~17:00)

受講料(税込)

550,000円

- ※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。
- ※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対象者

ソフトウェア開発に初めて携わる方

前 提 条 件

特になし

到達目標

- 個人のプログラミングと企業におけるソフトウェア開発との違いを説明することができる
- ソフトウェア開発はどのように進めていけばよいのか説明することができる
- 各活動の目的/役割、必要性を説明することができる
- 上記に関連してとりあげた基本概念を説明することができる

ソフトウェアテスト入門講座



—ソフトウェアテストの基本を身につける—

ホワイトボックスやブラックボックステスト、単体・統合・システムテスト。ソフトウェア開発に携わるエンジニアなら、テストに何らかの形で関与されているでしょう。

これまでテストといえば、ウォーターフォール型開発プロセスで下流工程と呼ばれ、実装の後の作業とされてきました。このため、実装を主業務とするエンジニアの中には、テストによるソフトウェアの検証を疎かにするという傾向がありました。ここ最近では状況が変わり、テストファーストやテスト戦略、テスト設計などの考え方が浸透しつつあり、テストの重要性は広く理解されるようになりつつあります。しかし、テストの大切さを認識したとしても、日々多忙な開発業務に追われ、テストの基礎を学ぶ機会を得るのが難しいことや、テストを体系的に教育・指導できるエンジニアが少ないことは、多くの開発現場における悩みではないでしょうか。

このような悩みにお答えすべく、2日間という短期間で、基本的なテストの考え方や技法を効果的かつ効率的に身に付けていただけるセミナーを豆蔵で開発しました。本講義で得た知識は、テスト設計演習を通して開発現場ですぐに実践していただける内容となっています。テストや開発現場での経験豊富なコンサルタントが作成した、ソフトウェアエンジニアリングに基づいた本セミナーは、テストの基本を一からしっかりと身に付けたいエンジニアにとって、最適なコースです。

本コースでは、御社へコンサルタントが同って実施するオンラインセミナー、弊社にて会場をご用意するオフサイトセミナーをそれぞれご用意しております。皆様のご受講をお待ちしております。

日 数

2日

時 間

6.5時間/日(10:00~17:30)

受講料(税込)

1,100,000円

- ※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。
- ※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対象者

システム開発、ソフトウェア開発、ソフトウェアテストに携わる方

前 提 条 件

ソフトウェアの開発経験

到達目標

- ソフトウェアテストについての基礎知識を習得し、実践できるようになる
- 自分のプロジェクトでテスト設計を行えるようになる

内 容

- はじめに
- 企業におけるソフトウェア開発の特徴
- エンジニアリングとは
 - エンジニアリングとは
 - エンジニアリングの構成要素
 - 主要な体系
 - 本講座でとりあげる内容
- プロジェクト立ち上げ時の活動
- 要求エンジニアリング(狭義)の活動
 - システム要求の獲得
 - システム要求分析
 - システム要件/仕様の作成
 - 要求エンジニアリングの重要性
- ソフトウェアエンジニアリングの基本活動
 - ソフトウェア設計
 - ソフトウェア構築
 - ソフトウェアテスト
 - レビュー
 - 妥当性確認
- 開発の進め方
- 管理・支援活動の(超)概要
 - 要件管理
 - 計画
 - 進捗管理
 - 構成管理
 - 品質保証

内 容

- ソフトウェアテストの基礎
 - テストとは
 - テストの目的
 - テストの7原則
 - テストの心理学
- 開発ライフサイクルとテスト
 - ソフトウェア開発プロセス
 - 典型的なテストレベル
 - テストレベルの意義
- 動的テストと静的テスト
 - 動的テスト
 - 動的テストとは
 - 機能テスト
 - 非機能テスト
 - 構造テスト
 - 変更部分のテスト
 - 静的テスト
 - 静的テストとは
 - レビュー
 - 静的解析(コードレビュー)
- テスト設計技法
 - 演習① Myersの三角形
 - テスト設計技法のカテゴリ
 - 仕様ベースの技法
 - 演習② 境界値テスト
 - 演習③ デシジョンテーブルテスト
 - 演習④ 状態遷移テスト
 - 演習⑤ 1. 構造ベースの技法
 - 演習⑤-1 スタートメントテスト
 - 演習⑤-2 デシジョンテスト
 - 演習④ (参考)経験ベースの技法
- テスト設計演習
 - 演習⑥ コンポーネントテスト設計
 - 演習⑥-1 モジュールのテスト
 - 演習⑥-2 制御パステスト
 - 演習⑥-3 テストデータ
 - 演習⑦ 統合テスト設計
 - 演習⑦ 統合テスト設計
 - 演習⑧ システムテスト設計
 - 演習⑧ システムテスト設計
- テストプロセス
 - 演習⑨ テストプロセスとは
 - 演習⑨-1 テストの計画と管理
 - 演習⑨-2 テスト分析・設計
 - 演習⑨-3 テスト実装・実行
 - 演習⑨-4 不具合の報告と管理
 - 演習⑨-5 不具合報告書の作成
 - 演習⑨-6 テスト報告・終了作業
- その他の話題
 - 演習⑩ ツールによるテストの効率化
 - 演習⑩ リスクとテスト

エンジニアリング基礎講座

—ソフトウェアorシステムエンジニアリングの基本を理解する—

UML やオブジェクト指向等の教育はいろいろあります。しかし、UMLやオブジェクト指向はなんのためにあるのでしょうか?モデルが読めたり、モデルが描けたりすることのためにあるのでしょうか。いいえ、UML やオブジェクト指向はあくまでも道具です。目的は、エンジニアリング力を上げることです。具体的には、思ったもの/アイデア/よいものを表現できる力、また、それをより高い品質・生産性ですばやく世の中に出すことができる力です。

どの世界でも基礎が大事です。基礎の分しか上には伸びていきません。プロのエンジニアこそ基礎/基本を大事にしています。基礎を理解している人は道具を有効に使いこなすことができます。

本コースの目的は、オブジェクト指向や構造化設計といった道具を更に上のレベルで使いこなし、エンジニアリング力をアップさせるために必要な基礎を習得することです。例えば、3〜5 年間オブジェクト指向を使っているが、更にステップアップを目指したいと思っている人、CMM を使って本当にエンジニアリング力をアップさせたいと考えている人、あるいは CMM レベル 3 のエンジニアリングとは具体的にどうなるのか模索している人には最適なコースです。(本コースは CMM レベル 3 の要件を満足する内容になります)

プロセス、技術を統合し、新しいエンジニアリング・ステージにアプローズする豆蔵にしかできない教育となっています。

さらに、本コースには、ソフトウェア開発経験のないマネージャや経営層を対象とした「**マネージャ向け エンジニアリング基礎講座**」もご用意しています。開発の難しさやソフトウェアエンジニアリングの全体像を短時間で理解できる内容となっており、要求・設計・テストの要点を押さえながら、内製化を進める上でポイントを効率的に学ぶことができます。

日 数

1 日

時 間

7 時間(9:00~17:00)、マネージャ向け 4 時間(13:00~17:00)

受 講 料 (税 込)

550,000 円

- ※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
- ※オンライン・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)ともに同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

システム開発、ソフトウェア開発に携わる人

前 提 条 件

ソフトウェアの開発経験者

到達 目 標

- ソフトウェアあるいはシステムエンジニアリングを構成する各活動の目的/役割、主な実施事項、作業成果物が理解できるようになります。特に分析の役割、設計との違いがわかるようになる。
- 開発プロセスおよび上記各活動との関係を理解できる。
- 要求、要件、仕様、モデルの違いを理解することができる。

要求基礎講座

システム開発・ソフトウェア開発において、発生するエラーの主な原因は、要求・仕様の不完全さだと言われています。実際、不正確な要求による開発の混乱やコードの書き直しといった手戻りを体験されている方も多いと思います。

我々がコンサルティングで訪れた開発現場でも、どのように要求を獲得するのか、要求をどのようにして仕様まで落としこんでいくのかといった課題はどの現場でも常に存在しており、システム開発・ソフトウェア開発を改善していくための普遍的な課題のひとつといえます。

本講座はそうした「要求」とそれに関係する概念を体系的に整理し、要求とは何か、要件とは何か、また、システム開発・ソフトウェア開発におけるそれらの役割について解説していきます。また、要求や要件を他者に伝えたり、他者と共有する際のポイントや内容を確認するためのポイントといった開発の現場で役立つノウハウや、ユースケースや形式手法といった要求と関連の深い技術についても紹介していきます。

こうした要求に関する概念を体系立てて整理している講座は大変珍しいものです。ですので、本講座は、これから開発の世界に入ってくる人が基本的な知識を身につけるためにも役に立ちますし、日頃要求に悩まされている要求管理担当の方がその原因を発見するのに役立ちます。是非、受講してみてください。

日 数

1 日

時 間

7 時間(9:00~17:00)

受 講 料 (税 込)

550,000 円

- ※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
- ※オンライン・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)ともに同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

システム開発者、ソフトウェア開発者、要求管理担当者、製品企画担当者、システムを発注する側の要求担当者

前 提 条 件

特になし

到達 目 標

- 以下の重要な項目をよく理解し、人に説明できるようにする
 - ・ 要求や要件とは何か(目的/役割、要点、互いの関係、実践のポイント)
 - ・ 要求や要件に対する分析の役割
 - ・ 要求や要件が備えるべき特性とその精査方法のポイント
 - ・ 要件の伝達・共有のポイント
 - ・ 妥当性確認の必要性・プロトタイプングの留意点・実施のポイント
 - ・ 各種レビューの存在と使い分け
 - ・ 戦略的な要求・要件開発の重要性と実践方法
- 要求や要件に関し、自組織の弱いところを識別でき、改善の方向性を提示できるようになる

内 容

- はじめに
- エンジニアリングとは
- エンジニアリング活動の全体構成
- プロジェクト発足時の活動
 - 1)プロジェクト発足時の活動の流れ
 - 2)商品/システム企画立案
 - 3)システム構想設計
 - 4)演習
- システム要件分析&システム設計
 - 1)システム要件分析&システム設計の流れ
 - 2)システム要求の獲得
 - 3)システム要求分析
 - 4)システム要件の作成
 - ・いわゆる"システム要求"とは？
 - 5)システム設計
 - 6)妥当性評価
 - 7)演習
- サブシステム分析・設計
- ソフトウェアの設計・実装
 - 1)ソフトウェアの設計・実装の流れ
 - 2)ソフトウェアの設計・実装で重要なこと
 - 3)ソフトウェアモジュールの設計
 - 4)実装
 - 5)演習
- テスト
 - 1)ソフトウェアテスト、システムテストの流れ
 - 2)テストの進め方
 - 3)各テストの目的
 - 4)全体テスト設計・計画
 - ・テスト/品質保証のタイプ
 - 5)テスト分析・設計
 - ・テスト関連作業成果物の整理
 - 6) テスト計画
 - 7)テスト詳細設計・実装
 - 8) テスト環境の準備
 - 9)テスト実施
 - 10)テスト報告
 - 11)テストの検証項目
 - 12)演習
- 後工程
 - 1)後工程の流れ
 - 2)支援文書の作成
 - 3)ポストモータム
- 開発プロセスの設計
プロジェクトにおいて、実際にこれらの活動をどう進めるかのプロセス設計に関して理解します。
 - 1)開発プロセス設計のパラメータ
 - 2)マイクロレベルのプロセス
 - 3)マクロレベルのプロセス
 - 4)開発プロセスの設計と計画との関係
 - 5)演習

内 容

- はじめに
- 要求の重要性と問題
 - 1) エラーの主原因
 - 2) よく目にする好ましくない現象
 - 3) 要求・要件・仕様一般に認知されている問題点
- 要求領域の概要
 - 1) 要求領域～本講座で取り上げる範囲～
 - 2) 要求エンジニアリング(狭義)の活動構成
 - 3) プロジェクト立ち上げ時の活動～先立つ活動～
 - 4) システム要求の獲得
 - 5) システム要求の分析
 - 6) システム要件/仕様の作成
 - 7) 活動間の関係のまとめ
 - 8) 要求・要件・仕様の違いと関係
 - 9) その他識別すべき違い
- 要求の獲得
 - 1) 要求とは何か
 - 2) 要求の種類
 - 3) 非機能要求(品質要求)
 - 4) 要求の構造
 - 5) 要求の獲得のステップ
 - 6) 要求の生成源の特定
 - 7) 要求の抽出
 - 8) 要求の折衝
 - 9) 要求の文書化
- 要求の分析
 - 1) 分析の概要
 - 2) シナリオ定義・分析
 - 3) モデルベース分析
 - 4) 要求・要件への抽出の利用
- 要求の作成
 - 1) 要件作成の概要
 - 2) 要件とは
 - 3) 要件と要件との関係
 - 4) 要件作成のステップ
 - 5) 要件の抽出
 - 6) 要件の文書化
 - 7) 要件の合意
 - 8) 要件と仕様
 - 9) 要求のもう一つの文書
- 要求・要件の事前精査
- 要件の伝達と共有
- 要求の妥当性確認
 - 1) 妥当性確認とは
 - 2) 妥当性の手法/技法
 - 3) 妥当性確認の基本ステップ
- 部分システムの場合の対応と要求の割り当て
- 要求管理
- 要求に関する計量
 - 1) 計量の典型的な視点/目的
 - 2) 代表的な計量尺度
- 要求開発
 - 1) 要求の特性
 - 2) 要求・要件の成長の側面
 - 3) 要求開発
- 要求ツール
 - 1) 要求ツールの主機能
 - 2) 代表的なツール
- まとめ

ソフトウェア設計基礎講座

ソフトウェア設計について、体系的に学んだことはあるでしょうか。ソフトウェア設計は、開発者であれば毎日行っていることでありますが、改めて、ソフトウェア設計とはどのようなもので、どのように考えていくべきものなのかと聞かれると答えに困ってしまう方も多いのではないのでしょうか。また、コンサルティングの現場でも、構造化設計やオブジェクト指向設計を適用してはいるが、その根底にあるソフトウェア設計の意味を理解しないまま実施しているという場面を多く目にします。

そうしたソフトウェア設計とは何か「が理解されていない状況では、いくら設計手法を適用しても、より良い方向に設計を改善していくことができないため、狙ったような効果は得られません。ソフトウェア設計の進め方・考え方には原則があり、それが身に付いていないと、どんな設計手法も有効に使うことができないのです。

本講座はそうしたソフトウェア設計とは何か、良いソフトウェア設計とはどういうことなのかを理解するための講座です。本講座を受講することで、ソフトウェア設計の基本を体系的に、かつ、設計手法に依存しない形で習得できます。

これからソフトウェア開発者になるという人、ソフトウェア設計のレベルアップを図りたい人、自身の設計判断に裏付けの欲しい人等々、初心者からベテランまで経験に応じて得られるものがある講座です。

もう設計者として仕事をしているという方も、この講座でもう一度基礎からやり直してみませんか。

日 数

1 日
2 日

時 間

1 日 7 時間(9:00～17:00)
2 日 6 時間/日(10:00～17:00)

受 講 料 (税 込)

1 日:550,000 円、2 日:1,100,000 円

※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

ソフトウェア開発者

前 提 条 件

製品/システム開発の経験があること(プログラミング言語の種類は問いません)

別 達 目 標

- 設計とは何かを理解する
 - 設計の目的・役割・活動の概要
 - 設計と前後の活動(ex. 要求分析)との関係
 - 設計原則の重要性和各原則の意味
 - 設計品質の意味とその適用
 - 設計の進化の必要性和手段
- 設計に関し自組織/自身の弱いところを識別し、学んだことを反映できるようになる
 - 設計課題の識別ができるようになっていくこと
 - 設計原則の設計解への適用ができるようになっていくこと
 - 設計解の設計品質が評価できるようになっていくこと
 - 設計解とソフトウェア構造の違いが分かっていること

プログラミング基礎講座

— 本物のプログラマーになるための基礎知識と作法の習得 —

システム開発のライフサイクル中で、いわゆる「プログラマー」が担当している作業は意外と多いものです。なぜか「プログラマー」は既に作成されている詳細設計に従って)単にコーディングだけを実施している」といった誤ったイメージが広がってしまったため、多くのプロジェクトでは比較的经验の浅い(若手)新人技術者がプログラマーとしてアサインされ、ソフトウェア構築(ソフトウェア・コンストラクション)の作業を担当しています。しかし、実際にプログラマーが担当する作業には、コーディングの他に、ルーチン詳細設計、開発者テスト、デバッグ、リファクタリング、チューニング、等々、高い品質意識と経験が求められる高難度の作業が多く含まれています。経験の浅い若手技術者は、それらの作業/作法を習得しながら(翻訳されつつ)開発を進めますが、効率の良い手順/良い作法がしっかり修得される前に試行錯誤的に多くのソース・コードが作成されてしまうこととなります。しかも、その際に手本として参照されるのが、先輩プログラマー(が)やはり若手の時代に)試行錯誤的に書いた(決して作法が良いとは言えない)既存ソース・コードであったりすると、組織/チームとして良いプログラミング作法の定着/ソース・コード品質の向上がなかなか進まないどころか、悪い作法/品質低下が広がってしまうマイナス・スパイラルな状況に陥ってしまうこともあります。このような状況の組織/チームでは、どこかで作業改善/意識変革を行わなければなりません。

本講座は『プログラミング基礎講座』という名称になっていますが、その実質は『ソフトウェア構築基礎講座』とも言えるような内容になっています。特定のプログラミング言語の構文や言語機能を解説するよくある「プログラミング入門」的な講座とは異なり、本講座では、特定のプログラミング言語によらないソフトウェア構築の基本的な考え方、要点、進め方、原則/ガイドラインといった基礎/作法の習得とソース・コード品質に対する強い意識付けにフォーカスします。ソフトウェア構築全般のうち、コーディング、デバッグを中心としたプログラマーが担当する主要なアクティビティについて、演習を交えながら習得していきます。

本講座を受講することによって、高品質のソース・コードを高い効率で生産できるようになるためのいどぐを掴むことができます。特に、真の(プロの)プログラマーとして自立していくための最初の一步として最適です。

日 数

1 日
2 日

時 間

1 日 7 時間(9:00～17:00)
2 日 7 時間/日(9:00～17:00)

受 講 料 (税 込)

1 日:550,000 円、2 日:1,100,000 円

※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- プログラマーとしてソフトウェア構築フェーズの作業を担おうとしている人/既に担当している方。
- プロのプログラマーとしての高い品質意識と良い作法を身につけたい中堅技術者/新人技術者。
- ソース・コードの品質チェックをする役割を持つ人/レビュー担当者。

前 提 条 件

- 何らかのプログラミング言語による実装経験があること(講座受講レベルでも可)。
- できれば、Smalltalk, C++, Java, C#, Ruby などのオブジェクト指向言語の知見があるとベター。

内 容

形式:講義+演習(個人)

1. はじめに

- 1) 本講座の目的
- 2) 到達目標
- 3) 前提条件
- 4) 進行の目安

2. 設計とは

- 1) ソフトウェアエンジニアリングの基本活動
- 2) なぜ設計が必要なのか
- 3) ソフトウェア設計の目的・実施事項
- 4) 問題・課題の識別と特定
- 5) 解の考案と選定
- 6) ソフトウェア構造の決定と記述
- 7) ソフトウェア設計階層
- 8) 前後のプロセスとの関係と役割

3. 問題・課題の識別と特定

- 1) 問題・課題の識別の進め方
- 2) ソフトウェア設計における主要な課題

4. 解の考案と選定

- 1) 解の考案と選定の進め方
- 2) ソフトウェアの設計原則
- 3) ソフトウェア設計の品質
- 4) 解の選定

5. ソフトウェア構造の決定と記述

- 1) ソフトウェア構造の決定と記述の進め方
- 2) 代表的な記述方法
- 3) ソフトウェア設計書の構成

6. 派生開発時のソフトウェア設計

- 1) 派生開発時の設計手順
- 2) 派生開発時に残す情報

7. ソフトウェア設計戦略

- 1) ソフトウェア設計戦略とは
- 2) 事前に特定すること
- 3) ソフトウェア設計戦略で決定すること

別 達 目 標

- プログラムに対する高い品質意識を持つようになること。
- ソフトウェア構築に関して(プロジェクト/自組織)の弱いところを識別でき、改善の方向性を提示できるようになること。
- ソフトウェア構築に関連するおもな概念、原則、手法、技法の概要を把握し、必要に応じて自分でさらに詳しく調べていくようになること。
- 各種の注意点やガイドラインを単に暗記するのではなく、それらの存在理由を理解すること。

内 容

形式:講義+演習(個人/グループ)

1. はじめに

2. ソフトウェア構築とは

3. プログラム品質

4. チーム開発

5. コーディング

- 5.1 型の利用
- 5.2 変数の利用
- 5.3 制御構造の適用
- 5.4 名前付け
- 5.5 コメント
- 5.6 防衛的プログラミング
- 5.7 エラー/例外処理
- 5.8 コーディング規約

6. テスティング

7. デバッグ

8. 最適化

9. リファクタリング

10. おわりに

付録

設計原則入門基礎講座

ー 拡張性や再利用性のための基礎知識と設計力の向上 ー

オブジェクト指向設計のメリットである「拡張性」や「再利用性」を享受するために考えるべき、11 の設計原則を学びます。個別の設計原則を学んだ後、設計原則を複数適用して設計をする総合演習を実施します。このことにより、実践で活かすことのできるスキルを身に付けることができます。

構造化手法からオブジェクト指向設計への遷移により拡張性や再利用性が高められたと言われます。しかし世の中の多くのシステム開発では、本当にこの拡張性・再利用性を享受できているのでしょうか。

本講座では、オブジェクト指向設計で失敗(拡張・再利用しづらい設計に)しないために、設計に携わる人にとって必須の知識である設計の原則を学ぶことができます。

本講座は 2 日構成となっており、1 日目は基本的な設計原則、2 日目はパッケージの設計原則と総合演習を扱います。1 日目のみを開催することも可能です。

また、Java・C++・C#といった主要な言語環境に対応しており、受講者の開発現場に即した形で学ぶことができます。

日 数

1 日
2 日

時 間

1 日 6.5 時間(10:00~17:30)
2 日 6.5 時間/日(10:00~17:30)

受 講 料 (税 込)

1 日:550,000 円、2 日:1,100,000 円

※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)とも同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- システム開発経験 2~3 年目で、設計者にステップアップしたいプログラマー
- オブジェクト指向設計を学習したい設計者
- オブジェクト指向を使っているにも関わらず、特性を活かしきれていない設計者

前 提 条 件

- クラス、継承、インターフェイス、ポリモーフィズムなどオブジェクト指向の概念・用語を理解していること
- UML(特にクラス図)を使ったコミュニケーションが取れること
- オブジェクト指向開発の流れを理解していること
- 弊社トレーニング「オブジェクト指向分析・設計演習」修了程度の知識

訓 練 目 標

- 再利用性や変更容易性を確保した設計ができる
- 各原則の特徴とメリットを説明できる
- 再利用性や変更容易性を低下させる設計を見分けられる
- 設計原則を適用すべきかどうかを判断できる
- 設計結果の根拠が説明できる

ソフトウェアテスト基礎講座

ー テストを効率よく実施し品質を確保するための基本知識、スキルを習得するー

ソフトウェアの大規模化、複雑化に伴い、ソフトウェアテストに要する期間・工数も増大し、開発に占めるテスト割合が大きくなっていきます。一方でソフトウェアの開発期間は短縮化の方向に進んでいるためにテストによる品質確保が十分に出来ず、結果として不具合が市場に流出し問題となる場合が増えていきます。いかにしてテストを効率よく実施し品質を確保するかは、ソフトウェア or システム開発の大きな課題であると言えるでしょう。

このようなソフトウェアテストが抱える課題に対して、近年、ソフトウェアテストの考え方は、より構造的な方向へ進化をきています。テストはもはやコーディングの後から始まる一つの活動ではなく、開発ライフサイクル全体を通して考えるもの、つまり要求プロセスの初期段階から始動し開発の進行に合わせて詳細化していくものだととらえるようになっていきます。

しかしながら、現状のソフトウェアテストに関する講座の多くは、開発ライフサイクル全体を通したテストの考え方を具体的に解説するまでには至らず、方法論や概念の解説に留まってしまっています。これに対し、本講座は、各活動の実施事項や作業成果物の説明、全体テスト計画立案や、テスト設計技法を習得するための演習問題など、学んだ内容を即現場にて実践できるようにしています。

本講座は以下のように 2 日構成となっています。

コース名	特徴	所要日数
第 I 部 プロセス編	ソフトウェアテストの全体構成と、各活動の目的と進め方を具体例と演習を通して学ぶ。	1 日
第 II 部 技法編	基本的なテスト設計技法を体系的に紹介し、演習を通してソフトウェアテスト設計スキルを習得する。	1 日

また、I 部、II 部で学んだ内容を実施するために実在するドキュメントを題材にしたワークショップ(テスト計画、テスト分析、テスト設計、テストケース設計技法)もご要望に応じて実施しております。

ソフトウェアテストの全体構成を知りたい方、テスト設計技法の基本を知りたい方のみならず、開発ライフサイクル全体を通したテストは具体的にどうなるか模索している方にも最適なコースとなっております。テストでお悩みの皆さん、是非本講座をご活用ください。

日 数

第 I 部 プロセス編 : 1 日
第 II 部 技法編 : 1 日

時 間

第 I 部 プロセス編 : 6.5 時間(10:00~17:30)
第 II 部 技法編 : 6 時間(10:00~17:00)

受 講 料 (税 込)

第 I 部 プロセス編 : 550,000 円
第 II 部 技法編 : 550,000 円

※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)とも同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

【プロセス編】
プロジェクトマネージャ/リーダー、アーキテクト、テストマネージャ、QA マネージャ、テスト設計担当者、開発者、プロセス改善推進者

【技法編】
アーキテクト、開発者、テスト設計担当者

内 容

形式:講義+演習(個人)

【1 日目】

1.ソフトウェア設計における問題

悪い設計とその設計によってどのような影響がでるのかを知り、なぜそのような設計になってしまったのかを学びます。

- 1) 設計を改善するためのオブジェクト指向
- 2) オブジェクト指向設計が生み出す新たな問題

2.変更に強いクラス構造を設計する

クラス、インターフェイスの凝集度を高め、結合度を下げるための原則を学びます。

- 1) 変更に強いクラス構造を設計するには
- 2) 単一責任の原則
- 3) 開放/閉鎖の原則
- 4) リスコフの置換原則
- 5) 依存関係逆転の原則
- 6) インターフェイス分離の原則
- 7) クラス構造の設計原則間の関係性

【2 日目】

3.再利用性に強いパッケージ構造を設計する

パッケージの作成、相互作用、利用方法に関する原則を学びます。

- 1) 再利用しやすいパッケージ構造を設計するには
- 2) 再利用・リリース等価の原則
- 3) 全再利用の原則
- 4) 閉鎖性共通の原則
- 5) 非循環依存関係の原則
- 6) 安定依存の原則
- 7) パッケージ構造の活用ポイント

4.総合演習:クラス構造、パッケージ構造を適用する演習

クラス構造、パッケージ構造を適用する演習を通して講座内容を確認することができます。

前 提 条 件

【プロセス編】

ソフトウェアの開発経験があること。

【技法編】

ソフトウェアの開発経験があること。

訓 練 目 標

【プロセス編】

- ソフトウェアテストの用語等基礎知識項目の意味を人に説明できるようになる
- 現状の一連のテスト活動における要強化ポイント・改善のポイントを識別、説明できるようになる
- テスト戦略/テスト全体計画が作成できるようになる

【技法編】

- ソフトウェアテストで使う代表的なテストケース設計技法について
 - 概要を理解する。
 - 活用方法を説明できる。
 - 現場で実践できる。

内 容

<< 第 I 部 プロセス編 >>

1. はじめに
2. ソフトウェアテストの最近の動向
3. ソフトウェアテストの基礎
 - 品質領域における位置づけ、重要な性質、限界、視点、基本用語
 - 機能不全に関する用語、テストとデバッグの違い
4. テストレベル
 - 典型的なテストレベル、ホワイトボックステストとブラックボックステスト、テストレベルと開発プロセス
5. テストの種類
 - 構造テスト、機能テスト、非機能テスト、変更部分のテスト、
6. テストケース設計(演習 2 問)
 - 全体テスト設計・計画、テスト分析・設計、テスト計画、テスト詳細設計・実装、テスト環境準備
 - テスト実施、テスト報告
7. ソフトウェアテストツール
 - テストジェネレータ、テスト実行フレームワーク、テスト評価ツール、テストマネジメントツール、性能分析ツール
8. テストを成功に導くために

<< 第 II 部 技法編 >>

1. はじめに
2. テスト活動の全体構成
 - 1) テストの各活動の概要
 - 2) テスト分析・設計
 - 3) テスト詳細設計・実装の流れ
3. テストケース設計
 - 1) テストケース設計の概要
 - 2) テストケース設計の実施事項
4. テスト技法
 - 1) テスト技法の概要
 - 2) 網羅を主眼にした技法
 - 3) 少なくすることを主眼にした技法
 - 4) 欠陥発見を主眼とした方法
5. テスト技法と演習
 - 1) 同値分割と境界値分析(演習 2 問)
 - 2) ドメインテスト(演習 1 問)
 - 3) デジモンテーブル(演習 3 問)
 - 4) 原因結果グラフ(演習 2 問)
 - 5) ペアワイズテスト(演習 2 問)
 - 6) 制御パステスト(演習 1 問)
 - 7) 状態遷移テスト(演習 1 問)

テストプロセスの進め方を体験し、実践へのヒントを得る

ソフトウェアテスト基礎第Ⅰ部(基本編:テストプロセス、全体テスト計画に焦点)、第Ⅱ部(テスト技法に焦点)で学んだ内容を“実践”する講座です。
ソフトウェアテスト基礎講座は、内容・構成とも実践を意識したものになっていますが、体験による実感や気づきを得る機会として、本ワークショップを提案いたします。
参加者による体験学習を重視するため、講義は必要最低限に留めています。事前にソフトウェアテスト基礎を受講されることが推奨されます。

テスト基礎ワークショップは、テーマに応じて以下のコースに分かれています。

コース名	特徴	所要日数
計画編	ソフトウェアテストプロセスでも重要な、テスト計画の考え方を、具体的な題材を用いた演習で体得します。 ソフトウェアテスト基礎第Ⅰ部(全体テスト計画)に対応しています。	1日
設計編	効果的・効率的なテストに欠かせないテスト分析・テスト設計の進め方を、具体的な題材を用いた演習を通じて考えます。 ソフトウェアテスト基礎第Ⅰ部(テスト分析・設計)、第Ⅱ部(テストケース設計技法)に対応しています。	1日

- 以下の特徴があります。
- 実際の製品に近い題材を取り上げること、ワークショップ形式の演習により、テストプロセスをどのように実践するのがよいか体感していただけです
 - 体験を振り返ることで、現在の業務(テスト計画、テスト設計)を再評価するきっかけが得られます(よい点、改善点等)
 - 演習題材にはお客様の実際の製品(成果物)を用いることも可能です。自分たちが実際に開発に携わる(携わった)製品を取り上げること、テスト基礎の内容をより実感できる効果があります。お気軽にご相談ください

現場でのテスト活動を改善したいと考えておられる方、学卒中心の研修では物足りないという方に適したコースとなっております。ぜひ本ワークショップをご検討ください。

日	教
■ テスト基礎ワークショップ 計画編	: 1日
■ テスト基礎ワークショップ 設計編	: 1日

時	間
■ テスト基礎ワークショップ 計画編	: 7時間 (9:00~17:00)
■ テスト基礎ワークショップ 設計編	: 7時間 (9:00~17:00)

受 講 料 (税 込)
■ テスト基礎ワークショップ 計画編 : 550,000円
■ テスト基礎ワークショップ 設計編 : 550,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。
※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)とも同料金です。
※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者
【テスト基礎ワークショップ 計画編】 プロジェクトマネージャ/リーダー、アーキテクト、テストマネージャ、QA マネージャ、テスト設計担当者、開発者、プロセス改善推進者
【テスト基礎ワークショップ 設計編】 アーキテクト、開発者、テスト設計担当者

静的解析・分析入門講座

静的解析ツールを使ったことはあるでしょうか？以前は、解析に時間がかかり、解析結果も誤検出が多いため、実プロジェクトで利用するには難しいツールでしたが、近年の解析技術の向上と、PCの性能向上とで、現在の静的解析ツールは品質保証の手段のひとつとして、十分使える状態に進化しています。また、そうした進化を受け、ISO26262他、静的解析ツールの適用を義務付けている機能安全規格も出てきています。
本講座はそうした静的解析ツールを、開発の中でどのように活用していけば良いのかについて取り上げた講座です。静的解析ツールでどのような問題を検出できるのか、静的解析結果をどのように分析すれば良いのか、静的解析ツールで指摘される問題点にどのように対応すれば良いのかといったことを実例を交えながら学んでいきます。また、代表的な静的解析ツールを紹介するので、ツール選定で迷っている方にも参考になると思います。
静的解析ツールから出力されたワーニングの多さに途方に暮れていた方、機能安全規格対応のために静的解析ツールを適用しなければならなくなった方々に最適な講座です。

日	教
1日	

時	間
6時間(10:00~17:00)	

受 講 料 (税 込)
550,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。
※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)とも同料金です。
※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者
静的解析ツールの導入を検討されている方

前 提 条 件
ソフトウェア開発の経験があること

例 達 目 標
■ 静的解析・分析とはどのような活動なのかを人に説明できるようになる
■ 静的解析でプログラムのどのような問題が検出できるのか理解できる
■ 静的解析結果を分析する際の着眼点について理解し、自身で分析できるようになる
■ 静的解析で発見される問題に対する典型的な解決策について理解し、自身で実施できるようになる

前 提 条 件
【テスト基礎ワークショップ 計画編】 ソフトウェアテスト基礎 第Ⅰ部 プロセス編を受講していること。
【テスト基礎ワークショップ 設計編】 ソフトウェアテスト基礎 第Ⅱ部 技法編を受講していること。

例 達 目 標
【テスト基礎ワークショップ 計画編】 <ul style="list-style-type: none">■ テストプロセス(テスト計画)の考え方・進め方を理解する■ 現在の業務で行なっていることの違いを認識する■ 現在の業務を再評価する(長所や改善点の発見)
【テスト基礎ワークショップ 設計編】 <ul style="list-style-type: none">■ テストプロセス(テスト分析・設計)の進め方を理解する■ テスト技法の使い方を体得する■ 現在の業務での進め方との違い/長所・短所を認識する概要を理解する

「テスト基礎ワークショップ 計画編」

- はじめに
 - 1.1 目的・目標
 - 1.2 前提条件
 - 1.3 講座の概要
 - 1.4 進捗の目安
- ソフトウェアテストの基礎
 - 2.1 テストとは
 - 2.2 テストの重要な性質
 - 2.3 テストの種類
 - 2.4 機能不全に関する用語
 - 2.5 テストの基本用語
 - 2.6 テストの視点
 - 2.7 テストの品質領域における位置づけ
 - 2.8 最近のテストの考え方
 - 2.9 テストレベル
 - 2.10 テストの種類
 - 2.11 テストプロセス
- テスト計画とは？
 - 3.1 テストにおける「計画」とは？
 - 3.2 IEEE Std 829とは
 - 3.3 テスト計画の要素
 - 3.4 テスト計画の枚層
- 全体テスト計画
 - 4.1 全体テスト計画での各要素概要
 - 4.2 全体テスト計画でのアプローチ作成までのステップ
 - 4.3 全体テスト計画でのアプローチまとめ
 - 4.4 演習問題を例に
- 個別テスト計画
 - 5.1 個別テスト計画での各要素概要
 - 5.2 個別テスト計画での目的・方針
 - 5.3 個別テスト計画でのスコープ
 - 5.4 個別テスト計画でのアプローチ
 - 5.5 アプローチの作成例(一部)
- 演習問題

「テスト基礎ワークショップ 設計編」

- テスト分析ワークショップの概要
 - 1.1 演習題材の概要
 - 1.2 今日のワークショップの前提用語
 - 1.3 進捗の目安
- テスト分析
 - 2.1 テスト分析・設計の概要
 - 2.2 テスト分析・設計の流れ
 - 2.3 テストベース分析
 - 2.4 仕様項目(テスト条件)の識別
- 演習問題 題材説明
 - 3.1 演習問題

内 容

- 形式: 講義 + 演習(個人)
- はじめに
 - 1) 目的と目標
 - 1) 前提条件
 - 1) 講義進行の目安
 - 静的解析・分析とは
 - 1) 静的解析とは
 - 1) 静的解析とは
 - 1) 静的解析・分析が求められている背景
 - 1) 静的解析関連用語
 - 1) 静的解析の種類
 - 1) 静的解析でできること・できないこと
 - 1) テストとの違い
 - 1) 静的解析・静的解析の効果
 - 1) 代表的な静的解析ツール
 - 静的解析で検出される問題
 - 1) コードの書き方に関する問題
 - 1) ソフトウェアメトリクスに関する問題
 - 1) ソフトウェア構造の問題
 - 1) セキュリティの問題
 - 1) オープンソースソフトウェアの問題
 - 静的解析結果の分析
 - 1) 分析の目的
 - 1) 分析の観点
 - 1) 分析の手順
 - 警告への典型的な対応
 - 1) 対応方法
 - 1) 警告への対応
 - 1) コーディング規約の対応
 - 1) メトリクスの改善
 - 1) 設計レビューの徹底
 - 1) 静的解析駆動開発
 - 静的解析の運用
 - 1) 運用時の典型的な課題
 - 1) 静的解析・分析の手順
 - 1) 開発プロセスへの統合
 - 1) CIへの統合
 - 1) 警告の管理

ソフトウェア品質保証入門講座

様々な製品に組み込まれ製品全体を制御したり、ユーザーに対するサービス提供の手段として使われたりなど、分野・形態を問わず、ソフトウェアの重要性は高まる一方であり、ソフトウェアの品質に対する要求もこれにつれて厳しくなっています。リリース後のソフトウェアトラブルが生じた時の影響の範囲や深刻さは今や見過ごすことのできないほどです。

このことはソフトウェアにおける品質保証活動の重要性が増していることを意味していますが、効果的なソフトウェア品質保証に苦勞している組織や、現状に疑問を感じている組織は今でも多く見られます。これには、①ソフトウェアが目に見えないものであるため、「ソフトウェア品質」というものも捉えどころがないと感じ、具体的な品質目標に落とし込めない、②ソフトウェア品質保証の適切な考え方を持っていないために、よい活動を構築できない、③ソフトウェア品質活動を展開し品質を確保しているつもりでの利害関係者から評価されていない、等の理由が考えられます。

本コースは、主としてこれからソフトウェア品質保証に取り組もうという方、既に実務を担当しているがソフトウェア品質保証に関する知識を整理したい方を対象に、ソフトウェア品質保証の考え方、基本的な概念、代表的な活動を学ぶことを目的とした講座です。業務としての品質保証ではなく、ソフトウェア開発の一部としての品質保証活動という切り口で整理しているので、テスト業務や品質保証業務の専任者に限らず、ソフトウェア開発担当者、アーキテクト、プロジェクトリダなど、ソフトウェアの開発・設計業務に関わる人全般にも適した内容となっております。

日 数

1日

時 間

6時間(10:00~17:00)

受講料(税込)

550,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- 品質保証を体系的に学びたいと考えている方
- 組織の品質保証活動構築を検討したい方

前 提 条 件

特になし(ソフトウェア開発経験か、テスト業務の経験があるのが望ましい)

例 達 目 標

- ソフトウェア品質保証の基本知識を説明できるようになる
 - ・ソフトウェア品質保証に関する基本概念(ソフトウェア品質、プロダクト品質、プロセス品質、品質要求、品質特性、V&V)
 - ・ソフトウェア品質保証計画の考え方

ソフトウェア構成管理基礎講座

「ソフトウェア構成管理」(以降 SCM と称します)は、CMM やソフトウェアプロダクトラインにおいて、ソフトウェア開発全体を支える基盤となるプロセスとして位置付けられており、その重要性が広く認知されてきています。製品開発サイクルの短期化やアジャイル開発の普及に伴い、その必要性は近年さらに高まっています。そして既に多くの開発現場に SCM ツールやプロセスが導入されています。

ところがここ数年、SCM の改善に関するコンサルティングの依頼が増えています。原因としては、以下のような要請から SCM が複雑化していることが挙げられます。

- 複数機種、複数仕向けといった多くのバリエーションの並行開発
- ソフトウェアの部品化やソフトウェアプロダクトラインへの取り込み
- ソースコードやドキュメントの組織的な共有化
- 開発拠点の分散化
- etc...

上のような課題を解決するために、高機能な SCM ツールを購入し、それに合わせてプロセスを修正するといった改善活動がよく行われますが、期待した効果が得られないケースが多く見受けられます。結果として、購入したツールの一部の機能だけが利用される、もしくは全く利用されなくなることも少なくありません。

また、近年は構成管理ツールである Git が広く普及していますが、単なるソフトウェアバージョン管理にしか使われていないケースが多く見られます。構成管理上のベースラインを設定せず、要件・設計・実装・テスト設計の食い違いを防げずに、プロジェクトが混乱に陥るケースも少なくありません。

そのような失敗を回避するためには、SCM とは何かをきちんと(実践的に基本を)理解する。現場の課題をきちんと識別・特定する。現場に則したプロセス・ツールを選択・導入する必要があります。しかし、残念ながら SCM に関する情報源は非常に少ないのが現状です。特に日本語では、書籍も数えるほどしかありません。CMM、プロダクトライン、SWEBOKにも SCM の説明がありますが、一般的な話ですので、現場に適用するにはそれなりのノウハウが必要になります。

そこで本講座では、SCM に関する重要な概念や活動内容等の基礎的な知識をしっかりと習得するとともに、実践上のポイント、支援ツールの選定作業、ソフトウェアプロダクトラインのような先進的な利用での留意点といった、実際の開発現場で役立つ知識についても習得します。少しでも多くの開発現場で SCM 活動が改善されることになれば幸いです。

日 数

1日

時 間

6時間(10:00~17:00)

受講料(税込)

550,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- 構成管理担当者、ビルド・インテグ担当者、開発インフラ・設備管理者、現場開発者、現場管理者、
- プロセス改善推進者、上位管理者

前 提 条 件

特になし

内 容

1. はじめに

- 1.1 本講座の目的と目標
- 1.2 進行の目安

2. ソフトウェア品質保証とは

- 2.1 ソフトウェア品質保証とは
- 2.2 ソフトウェア品質保証の範囲
- 2.3 ソフトウェア品質保証活動の分類

3. ソフトウェア品質保証に関する基本概念

- 3.1 ソフトウェア品質
- 3.2 プロダクト品質・プロセス品質
- 3.3 品質要求
- 3.4 品質特性

4. ソフトウェア品質保証の活動

- 4.1 ソフトウェア品質保証活動の全体構成
 - 4.2 ソフトウェア品質保証活動の基本概念
 - 4.3 品質保証計画
 - 4.4 検証
 - 4.5 妥当性確認
 - 4.6 不具合管理
 - 4.7 品質計量
 - 4.8 品質分析・評価
 - 4.9 原因分析・再発防止
 - 4.10 監査
 - 4.11 他の活動との関係
- ※ 途中に演習を挟みます

5. ソフトウェア品質保証を支える主な技術

- 5.1 レビュー
- 5.2 テスティング
- 5.3 静的検証(レビューを除く)
- 5.4 品質分析・評価
- 5.5 原因分析・再発防止

6. まとめ

例 達 目 標

- 以下を人に説明できるようになる
 - ・ 構成管理に関わる用語・概念
 - ・ 構成管理の目的/効果
 - ・ 構成管理の活動の全体構成
 - ・ 構成管理の各活動の目的・実施事項・要点
 - ・ 構成管理を支援するツールの機能概要と代表的なツール
 - ・ 複数製品・多バリエーション開発における構成管理の難しさ・課題・改善/対処のポイント
- 現在の構成管理の課題を洗い出し、具体的な改善活動が始められるようになる

内 容

1. はじめに

2. 構成管理とは
 - 1) 不適切な構成管理がもたらす典型的な問題
 - 2) 構成とは
 - 3) 構成管理を通して実現したいこと
 - 4) 構成管理の目的(まとめ)

3. 構成管理の基本概念

- 1) 構成
- 2) ベースライン
- 3) 構成 ふたたび
- 4) ライブラリ
- 5) バージョン・リビジョン
- 6) バリエーション
- 7) 基本概念間の関係

4. 構成管理の活動

- 1) SCM の全体構成
- 2) SCM 環境の構築/整備
- 3) ベースラインの確立
- 4) ベースラインの変更管理
- 5) ソフトウェアビルディング
- 6) ソフトウェアリリース
- 7) CI/CD 状況の記録

5. 構成管理の管理的な活動

- 1) SCM の計画
- 2) SCM 実態報告
- 3) SCM の監督
- 4) SCM の監査

6. 構成管理のツール

- 1) 構成管理ツールが備える機能
- 2) 構成管理ライブラリ機能の概要
- 3) 変更要求管理機能の概要
- 4) リリース機能の概要
- 5) 代表的なツール
- 6) 分散開発における構成管理システム
- 7) 自動ビルドツール

7. 複数製品間・多バリエーションの構成管理

- 1) 典型的事例
- 2) メインストリーム・プランチ
- 3) 複数製品間・多バリエーションの構成管理の課題と要点
- 4) 支援ツール例

8. まとめ

エンジニアリング技術紹介コース

一世の中にある代表的なエンジニアリング技術についてその技術の狙い/概要/効果を紹介する一

ソフトウェア危機が叫ばれ、25〜30年。以来、様々なエンジニアリング技術が考案されてきました。消えていった技術もありましたが、有力なものも蓄積されてきています。

ソフトウェアエンジニアリングを変えていくためには、それらの技術を有効に使うことが重要になり、目的にあった技術を選定あるいは組み合わせて使っていく必要があります。そのためには、世の中にどのような技術があるのか、それはどのようなものなのかを知らなければなりません。しかし、調査したり、勉強したりするには時間がかかりすぎ、開発に追われ、なかなか時間が取れないのが現状ではないでしょうか。また、現在、オブジェクト指向やテスト技法など特定のテーマに特化した教育は多々ありますが、今回提供するようなコースは見当たりません。

本コースは、現在使われている代表的なエンジニアリング技術を一同に集め、それらの適用範囲、概要、技術を使用する目的や狙い、考え方を理解することを目的とした、きわめてユニークなコースとなっています。これから、ソフトウェアエンジニアリングの改善を検討している人、最近のエンジニアリング技術の動向/概要を知りたい人、また新しいエンジニアリング技術を組織に導入したい人、あるいは組織のエンジニアリングを活性化させたいと思っている人には最適なコースです。

さらに、自動車分野に特化した「**エンジニアリング技術紹介コース(車載編)**」もご用意しています。モデルベース開発(MBD)、機能安全(ISO 26262)、AUTOSARといった車載固有の代表的技術を取り上げ、背景や目的、適用範囲を分かりやすく紹介します。車載システム開発に関わる方々にとって、技術動向を効率的に理解し、概観を捉えられるユニークなコースです。

ソフトウェアエンジニアリングを専門とする豆蔵ならではのコースとなっています。

日 数
1日
時 間
3時間(13:00~16:00) ~ 6.5時間(10:00~17:30)
受講料(税込)
343,200円~550,000円
※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。
※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。
※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。
対 象 者
システム開発、ソフトウェア開発に関わる人
前 提 条 件
■ ソフトウェアの開発経験があること ■ ソフトウェアのマネージメント経験があること
例 違 目 種
■ 代表的なエンジニアリング技術につき、その技術の狙い、概要、効果を理解する

- 2)ソフトウェア危機とは
 - 3)SysMLの特徴
 - 4)合意の種類
 - 5)SSMが提供するもの
 - 6)価値観の相違の例
 - 7)SSMによる効果
- 13. SysML (System Modeling Language)**
- 1)背景と課題
 - 2)SysMLとは
 - 3)ダイアグラム
 - 4)ダイアグラム:詳細
 - 5)効果
 - 6)注意点
- 14. Android**
- 1)背景と課題
 - 2)Androidとは
 - 3)アーキテクチャ
 - 4)効果
 - 5)注意点
- 15. OSGi**
- 1)背景
 - 2)コンポーネントと課題
 - 3)OSGiとは
 - 4)アーキテクチャ
 - 5)コンポーネント指向:バンドル
 - 6)サービス指向とSOA
 - 7)効果
 - 8)注意点
- 16. Scala**
- 1)背景と課題
 - 2)Scalaとは
 - 3)コード例
 - 4)効果
 - 5)注意点

車載編

- 1. 本講座の概要:目的と目標**
- 2. 車載システムの現在**
 - 1)今の車に求められているもの
 - 2)車載ソフトウェア開発の課題
- 3. 開発プロセス(Automotive SPICE)**
 - 1)背景と課題
 - 2)Automotive SPICEとは
 - 3)プロセスの全体像
 - 4)効果
- 4. デイペンダビリティ**
 - 1)背景と課題
 - 2)プロセスによるデペンダビリティの作り込み(バイ・デザイン)
 - 3)主な規格
 - 4)規格対応のポイント
 - 5)要求の生成源の違い
 - 6)安全を生み出すための設計
 - 7)検証の徹底
 - 8)安全であることの証明
- 5. MBD(Model-Based Development)**
 - 1)背景と課題
 - 2)MBDとは
 - 3)効果
 - 4)注意点
- 6. 形式手法**
 - 1)形式手法:背景/課題とるところ
 - 2)形式手法とは
 - 3)形式手法の種類と代表的手法
 - 4)例
 - 5)オブジェクト指向との相互乗り入れ
 - 6)形式手法:効果
- 7. MBSE (Model Based Systems Engineering)**
 - 1)背景と課題
 - 2)MBSEとは
 - 3)MBSE事例
 - 4)効果
- 8. SysML (System Modeling Language)**
 - 1)背景と課題

内 容

通常版(ソフトウェアエンジニアリング編)

- 1. 本講座の概要:目的と目標**
- オブジェクト指向(Object-Oriented: OO)**
 - 1)背景:ソフトウェア危機と構造化手法 構造化手法とは、構造化設計、構造化分析、構造化手法の効果、構造化手法の限界
 - 2)オブジェクト指向とは
 - 3)オブジェクト指向の基本思想 構造化との比較:カプセル化、継承、構造化との開発方法の違い
 - 4)オブジェクト指向による効果
- 3. コンポーネント/フレームワーク**
 - 1)背景
 - 2)コンポーネント/フレームワークとは
 - 3)フレームワーク:詳細
 - 4)コンポーネント:詳細
 - 5)CBD: Component Based Development
 - 6)コンポーネント/フレームワークを使った開発と効果
 - 7)コンポーネント:効果
- 4. UML(Unified Modeling Language)**
 - 1)UMLの背景
 - 2)UMLとは
 - 2)UMLで提供されている図
 - 3)UMLで提供されている図:詳細
 - 4)効果
 - 5)注意点
- 5. パターン**
 - 1)パターン:背景/課題とるところ
 - 2)パターンとは
 - 3)システム開発におけるパターンの種類と代表例
 - 4)デザインパターン
 - 5)アスペクト指向:詳細
 - 6)アスペクト指向:効果
- 6. アスペクト指向**
 - 1)アスペクト指向:背景/課題とるところ
 - 2)アスペクト指向とは
 - 3)アスペクト指向による効果
- 7. MDA**
 - 1)MDA:背景/課題とるところ
 - 2)MDAとは
 - 3)MDAの構造
 - 4)MDAのアプローチ
 - 5)CASEからMDAへ
 - 6)MDA対応ツールの紹介
 - 7)MDA導入による効果
- 8. プロダクトライン**
 - 1)プロダクトライン:背景/課題とるところ
 - 2)プロダクトラインとは
 - 3)プロダクトラインの基本的活動:管理、コア資産開発、製品開発
 - 4)プラクティスエリアとプラクティスパターン
 - 5)プロダクトライン:効果
- 9. ソフトウェアテスト**
 - 1)ソフトウェアテストの実際
 - 2)ソフトウェアテストの矛盾
 - 3)ソフトウェアテストの境界
 - 4)ソフトウェアテストの現実
 - 5)テスト技法の使い分け
 - 6)テスト技法の例
 - 7)ツールの活用
 - 8)テストの効果をあげるには
- 10. 開発プロセス:UP, XP**
 - 1)開発プロセス:背景/課題
 - 2)開発プロセス:UPとは
 - 3)開発プロセス:XPの背景
 - 4)開発プロセス(UP, XP)による効果
- 11. 形式手法**
 - 1)形式手法:背景/課題とるところ
 - 2)形式手法とは
 - 3)形式手法の種類と代表的手法
 - 4)例:Z, CafeOB, JSP
 - 5)オブジェクト指向との相互乗り入れ
 - 6)形式手法:効果
- 12. SSM:ソフトシステム方法論**
 - 1)SSM:背景/課題とるところ

- 2)SysMLとは
 - 3)ダイアグラム
 - 4)ダイアグラム:詳細
 - 5)効果
 - 6)注意点
- 9. 車載プラットフォーム**
- 1)背景と課題
 - 2)車載プラットフォームとは
 - 3)主な車載プラットフォーム
 - 4)効果
 - 5)注意点
- 10. プロダクトライン**
- 1)背景/課題とるところ
 - 2)プロダクトラインとは
 - 3)プロダクトラインの基本活動
 - 4)プラクティスエリアとプラクティスパターン
 - 5)プロダクトライン:効果
- 11. USDM**
- 1)背景と課題
 - 2)USDMとは
 - 3)USDMのフォーマット
 - 4)効果
- 12. トレーサビリティ**
- 1)背景と課題
 - 2)トレーサビリティとは
 - 3)トレーサビリティを実現する技術
 - 4)効果
- 13. アジャイル開発**
- 1)背景と課題
 - 2)アジャイル開発とは
 - 3)アジャイル開発の手法
- 14. まとめ**

CMM/CMMI 講座 ～レベル 2 SEPG 向け～

— CMMの基本概念、全体構造、レベル2の各プロセス領域の概要について理解する—

CMMは、国内でも適用が進められ、組織のプロセス改善や体質改善の有効なツールとして広く普及しています。CMMを使って改善を進めるには、まずはCMMを正しく、そしてよく、理解しておく必要があります。CMMの適用を間違え、思っているような改善効果は得られません。

現在、CMMをベースに社内プロセスを定義している企業は多く存在しますが、セミナーの開催は限定的で、新しい書籍もあまり出版されていません。CMMが難しいのは、その内容と現実とのギャップにあります。既存のセミナーや書籍の多くは、CMM自体の解説にとどまり、それだけではこのギャップは埋まりません。

本コースは、具体的な活動イメージを示しながら、CMMのポイントを解説していく、実践的なCMM教育コースです。豆蔵はこれまで多くの企業のCMMに基づく改善をお手伝いしてきました。そこでの経験、ノウハウを基に作ったのが、本コースです。

CMMとは何かを理解したい方、あるいはさせたい方、これからプロセス改善を進めようとして検討されている方に最適です。ぜひご受講ください。

日 数

1日

時 間

6時間(10:00～17:00)

受講料 (税込)

550,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対象者

プロセス改善推進者、現場開発者、現場管理者、上位管理者

前 提 条 件

ソフトウェアに関する基礎知識を持っている事

訓 達 目 標

- CMMの基本概念、全体構造、並びに各プロセス領域の目的が人に説明できるようになる。
- アセスメントの各弱み(所見)に関し、あるべき姿とともに、指摘内容が人に説明できるようになる。
- CMMの原テキスト(ex. TR-25)に基づき、自らプロセスの検討・定義が始められるようになる。
- CMMIの採用 or 利用に関し、具体的な検討が行えるようになる。

プロジェクト管理講座 基礎編

— 要件管理・計画・進捗管理 —

プロジェクトを成功させるのには何が必要か？ 固有技術も、エンジニアリングに関する技術も勿論不可欠です。ただそれだけではプロジェクトはうまくいきません。

開発は常に内外に不透明な要素を伴います。重要な/価値の高いプロジェクトほどそうです。また典型的には、供給者も含め、数多くの開発者が参加します。そういった中、プロジェクトの道筋を定め、プロジェクトに対する様々な外乱/変動の影響を制御しながら、リソースや活動を適切に配置・制御し、プロジェクトのゴールを達成、即ち成功に導くのがプロジェクト管理です。

プロジェクト管理のやり方もこれまでとは様変わりしています。個人的なkkd(勘と経験と度胸)に依拠した進め方は一昔前のやり方です。現在、プロジェクト管理は一つのエンジニアリング分野であり、きちんとした知識に基づいた合理的な活動であることが求められます。

現在、プロジェクト管理については、多くの書籍が出版され、多くのセミナー・教育講座も利用可能になっています。ただ、プロジェクトの個性もあって、それらの大半は、PMBOK(Project Management Body Of Knowledge)やCMMI(Capability Maturity Model Integration)そのもの、あるいはそれらをベースとした一般的な"知識"を教えるにとどまっているようです。

本講座は一般的な"知識"をお教えることは目的としていません。視点はあくまでも実践にあります。テーマも全てを網羅するのではなく、プロジェクト管理の中でも特に重要な、要件管理・計画・進捗管理(監視と制御)に絞り、各テーマ、一般的な"知識"を踏まえた上で、具体的な進め方・要点を活動ベースで解説します。弊社はこれまで多くの会社様の改善のお手伝いをさせていただいています。そのコンサルテーションの基本的なエッセンスを集めたものが本講座です。

本講座は既に、メーカー様を中心に多くの実施実績があります。本講座をベースに改善を図られ、大きな成果(ex. 品質向上・フロントーディング)を上げていらっしゃる会社様もあります。弊社の講座の中でも特にお薦めする講座の一つです。どうぞ、御社のエンジニアリング改善、メンバの育成にご活用ください。お役に立つはずですよ。

日 数

1日

時 間

6.5時間(10:00～17:30)

受講料 (税込)

550,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対象者

- 管理的業務を行なっている or 近く行なう予定/かもしれない開発者
ex. (サブ)グループリーダー、チームリーダー
- リーダの開発者
ex. アーキテクト
- プロジェクト管理者

内 容

1. はじめに

- 1)本講座の主な内容
- 2)本講座の構成
- 3)本講座を受けての達成目標
- 4)講義進行の目安

2. CMM 普及の背景と現状

- 1)現状～組込み系の場合～
- 2)周囲の取り組み状況
- 3)何故CMMなのか？

3. CMMの全体概要

- 1)CMMとは
- 2)CMMの基本思想
- 3)CMMのプロセスの定義
- 4)5段階からなる成熟度モデル
- 5)各成熟度のプロセスの状態
- 6)CMMの改善効果
- 7)CMMのモデル構造
- 8)各成熟度のKPA
- 9)ロコモフィーチャー
- 10)K&Pとゴール、成熟度の関係
- 11)IDEALモデル
- 12)CMMからの主要なメッセージ

4. レベル 2 の各プロセス領域の概要

- 1)レベル 2 プロジェクトのプロセス領域構成
- 2)要件管理(RM)
- 3)ソフトウェアプロジェクト計画(SPP)
- 4)ソフトウェアプロジェクト進捗管理(SPTO)
- 5)ソフトウェア構成管理(SCM)
- 6)ソフトウェア品質保証(SQA)
- 7)ソフトウェア外注管理(SSM)

5. CMMIについて

- 1)CMMIとは
- 2)CMMIにおける統合の狙い
- 3)CMMIの統合の過程/ベースモデル
- 4)SW-CMMからのモデル上の変更点
 - ・ 連続表現の追加、専門分野の導入、プロセス領域の構成の見直し、モデル構造の変更
 - 5)モデル内内容/記述面での変化

6. CMM 使用の留意点

7. 改善事例

前 提 条 件

- ソフトウェアの開発経験者
- 開発プロセスに関する基本的な知識を有していること

訓 達 目 標

- 要件管理・計画・進捗管理の各要点を習得し、自分自身で気づかないといけなところを明らかにする。

内 容

1. はじめに

2. 要件管理

- 1) 要件管理の目的
- 2) 企画構想からシステム要件/仕様作成までの基本的な流れ
- 3) 要望・要求・定義された要求・要件/仕様との関係
- 4) 管理すべき要件
- 5) 要望・要求の管理
- 6) 要件管理の基本
- 7) 演習

3. 計画

- 1) 計画の意義
- 2) 計画で取り上げられるべき項目
- 3) 作成する計画の種類
- 4) 計画作成の基本フロー
- 5) 計画に関する全般的な留意点
- 6) 演習

4. 進捗管理

- 1) 進捗管理の要点
- 2) 進捗管理を構成する活動
- 3) 演習

外注管理基礎講座

ここ最近の開発は、人材確保・開発期間短縮・開発コスト削減のために、外部の企業に開発を委託することが増えてきています。特に海外企業への委託や国内パートナーの活用が拡大しています。これに伴って自社開発あるいは社内での外部の企業と一緒に開発していた時には発生しなかったいつもの問題が発生してきています。

例えば、「計画通りに進捗としていると報告を受けていたが、外注先の成果物やレビューしたら大きな問題/課題があることがわかり、計画の見直しが必要になった」、「仕様を提示したにも関わらず、実装漏れや実装ミスが多く、手戻りが発生して納期に間に合いそうもないあるいは、「テストして初めて外注先の品質が悪いことに気がついた」といった問題を引き起こすことがあります。この主な原因としては、外注先の進捗状況確認、品質の確認が出来ていなかったり、あるいは、要求/仕様の出し方が悪かったり、受け入れテストが出来ていなかったりソフトウェア開発の実態にマッチした外注管理が実施されていないことが挙げられます。

本講座は、PMBOK(Project Management Body Of Knowledge)や CMMI(Capability Maturity Model Integration)といった一般的な“知識”を踏まえた上で、外注管理で“何を管理するのか”、“実施する上の要点は何か”を具体的な進め方・要点を活動ベースで解説します。弊社はこれまで多くの会社様の改善のお手伝いをさせていただいています。そのコンサルテーションの基本的なエッセンスを集めたものが本講座です。既にオフショア開発あるいは外部の企業に開発を委託しているが成果が出ていない方、あるいはこれから実施したいが進め方がわからない方にはお勧めの講座となっております。

日 数

1日

時 間

6時間(10:00~17:00)

受講料(税込)

550,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

ソフトウェアの外注管理に従事している方あるいは、これから外注管理を行う予定の方

前 提 条 件

開発プロセスに関する基本的な知識を有していること

例 達 目 標

- 以下の重要な項目をよく理解し、外注管理としてやらなければならないこと、気をつけなければならないことが明らかになる。
 - ・外注先の選定と評価
 - ・外注における要件管理、進捗管理、品質管理
 - ・外注におけるコミュニケーション
- 外注管理に関し、自組織の弱いところを識別でき、改善の方向性を提示できるようにする。

内 容

1. はじめに
2. ソフトウェア開発の外注管理
 - 1) よく見られる外注開発の問題点
 - 2) 外注とは
 - 3) 外注管理とは
 - 4) 外注管理プロセス
3. 調達管理
 - 1) 外注計画
 - 2) RFPの作成
 - 3) 外注先の選定
4. 契約管理
 - 1) 契約プロセス
 - 2) 契約交渉
 - 3) 契約締結
 - 4) 契約変更(訂正)
5. 開発管理
 - 1) 外注先開発計画の承認
 - 2) 進捗管理
 - 3) レビュー
 - 4) SQA活動監視
 - 5) 変更管理
 - 6) 是正措置
6. 検収
 - 1) 検収の目的
 - 2) 受入れ検証
 - 3) 成果物の移行
7. クロージング
 - 1) 目的
 - 2) プロジェクトの結果をレビュー
 - 3) 外注先の評価
 - 4) 今後の改善(支援)計画
8. コミュニケーション・スキル
 - 1) 外注管理とコミュニケーション
 - 2) コミュニケーションの特徴と基本原理
 - 3) コミュニケーションのコンテキスト
 - 4) コミュニケーション力
 - 5) 非言語によるコミュニケーションの重要性

OSS ライセンス管理基礎コース

ーオープンソースライセンス管理の基本を理解するー

PCやサーバに限らず、携帯電話やテレビ、DVDなどの身近な家電製品にもLinuxをはじめとするオープンソースソフトウェア(OSS)が広く使われるようになってきました。しかし、「オープンソースソフト」を「無料で自由に使ってよいソフトウェア」と誤解されていませんか? OSSには、高機能・高性能・高品質のソフトウェアが無料で利用できるというメリットがある一方で、周辺のソースコードの公開や特許の無償使用許諾を求められる場合があり、違反が発覚すれば訴訟や重大なコンプライアンス問題に発展するリスクもあります。そのため、OSSを正しく使いこなすためには、マネージメント層によるOSSライセンスに関する理解と、開発部門・管理部門におけるOSSコードの検出、独自開発ソフトとの分離など、ライセンス管理技術の向上が不可欠です。

さらに近年は、国際規格ISO/IEC 5230(OpenChain)が制定され、世界的にOSSライセンス管理の重要性が一層高まっています。コンプライアンス順守や取引先からの信頼確保の観点からも、組織として適切なライセンス管理が求められるようになりました。またOSS検出ツールの普及により、GPLの誕生した1980年代やLinuxが組込み分野に急速に普及した2000年代初頭と比べて、意図しないOSS流用の検出精度や効率は大きく向上しました。しかし一方で、検出結果から流用元のOSSを正確に特定するには依然として多大な時間を要するなど、新たな弊害も発生しています。

本コースの目的は、OSSライセンスの基礎知識と、OSS検出ツール導入時のプロセス構築の概要を知ることです。ソフトウェア組込み製品開発のマネージメントを担当される方でLinux、Android、TIZENなどのOSSをプラットフォームとして利用する製品を開発するにあたり、OSSライセンス管理の基礎知識を習得したい方、納入先へのOSSライセンス順守のエビデンス提出のためにOSS検出ツール導入と管理プロセス構築を担当される方などに最適なコースです。

OSSライセンス管理プロセスを効率よく構築するためのポイントをご説明する、企業の視点を重視した教育となっております。

日 数

1日

時 間

4時間(13:00~17:00)~6時間(10:00~17:00)

受講料(税込)

385,000円~550,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- ソフトウェア組込み製品開発のマネージメントを担当される方
- OSSライセンス管理プロセス構築をこれから担当される方、既にしている方

前 提 条 件

- 前提条件なし

例 達 目 標

- 主要OSSライセンスの概要と取扱を理解できる
- OSSライセンス管理技術の変化について理解する
- OSS検出ツール導入時の管理プロセス構築方法を理解できる

内 容

1. はじめに
2. 講習の位置付け
3. OSSライセンス概説
 - 1) オープンソースとは
 - 2) オープンソースの考え方
 - 3) 代表的なライセンス
 - 4) 最近の動向
 - 5) 訴訟
 - 6) OSS検出ツール
4. OSSライセンス管理プロセス構築のポイント
 - 1) OSSライセンス管理プロセス
 - 2) ツール導入時に発生しやすい問題
 - 3) ツール導入時の課題
 - ・情報量の増大
 - ・判定の複雑化
 - 4) ツール導入時の課題の解決
 - ・各ステップでの具体的な課題解決
 - ・契約の確認
 - ・調査
 - ・コンプライアンスギャップへの対処
 - ・ソースコード公開
 - 5) 導入から定常運用への移行
 - ・導入から定常運用への移行とは
 - ・商品開発プロセスとの連携
 - ・自社ソースコードの分離
 - 6) 教育
 - 7) 他部門との連携
 - 8) 情報伝達について
5. おわりに

ソフトウェアプロダクトライン入門講座

ますます厳しくなる製造業の世界においてマーケットが求める製品群をいかにタイムリーに高い生産性のもとで品質を確保してリリースするか、そのための体系的な取り組みが求められています。そのためには、単純にソフトウェア開発プロセスを導入するだけでは済まされません。

マーケットの求めるニーズに対応する製品企画と適切なソフトウェア工学にもとづく設計技術力と品質を確保した実装技術を備えた開発プロセスの実施、これらの戦略的な融合こそが求められているのです。そこでいま注目されているのがソフトウェアプロダクトラインの考え方です。この方法論はSEIで体系化され、製造業の組み込みソフトウェアやパッケージソフトウェアを中心にシリーズ製品開発を行うソフトウェア開発組織で導入する企業が年々増加してきています。

ビジネスとIT技術と組織の3者に戦略的な資産の再利用という観点で切り込むことにより、生産性や品質、納期を著しく改善する可能性に各社とも熱い視線を向けているのです。

豆蔵では、長年蓄積してきたオブジェクト指向技術、並びにCMM/CMMIの導入支援の成果をベースに、組織内で再利用を展開するプロダクトラインの導入に取り組んでいます。そこでの経験、ノウハウを基に作成したのが本講座です。

豆蔵の上級コンサルタントが例題の解説や受講者からの疑問にインタラクティブに答える形で、実際の適用や導入に関する知識も学んでいくことができます。

日 数

1日

時 間

7時間(9:00~17:00)

受講料(税込)

550,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※オンサイト・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対象者

- オブジェクト指向プログラミング言語を習得したい方
- 組み込み系ソフトウェアの現場の生産性や品質を向上させたいと考えているソフトウェア技術者の方
- 製造業において生産性と品質・納期の向上のためにソフトウェア開発プロセスを組織的に改善させたいマネージャの方
- 自分の事業や製品ドメインに対するソフトウェア開発技術やプロセスを組織的に見直したいマネージャや経営層の方

前 提 条 件

- 組み込みや製造業向けのソフトウェア開発の進め方の概要を理解されている方
- 組み込みや製造業向けではないが、シリーズ開発で組織的な再利用を検討されてきた方

到達目標

- ソフトウェアプロダクトラインの基本概念、全体構成、及びいくつかのプラクティスエリアの目的やプラクティスパターンを理解し、プロダクトラインの活動の価値を理解できるようになる。
- プロダクトラインにもとづくソフトウェア開発の方法論の概要を理解する。
- 適用事例から実際の取り組みを知ることによって、自社内で適用に関する検討開始の機会を掴めるようになる。

内 容

1. はじめに

- 1)本講座の主な内容
- 2)本講座の構成
- 3)本講座を受けての達成目標

2. プロダクトラインの背景と現状

- 1)現状 ~組み込み系の場合~。
- 2)プロダクトラインの背景・課題
- 3)周囲の取り組み状況
- 4)なぜプロダクトラインなのか？

3. プロダクトラインの概要

- 1)プロダクトラインとは？
- 2)プロダクトラインが目指す姿
- 3)プロダクトラインの全体構造
- 4)プロダクトラインの基本的活動

4. プロダクトラインのプラクティスエリア紹介

- 1)プラクティスエリア概論:29個に対する3つのカテゴリ
- 2)ソフトウェア工学
- 3)技術管理
- 4)組織管理

5. プロダクトラインのプラクティスパターンの紹介

- 0)プラクティスパターンの概論
 - 1)体系的な導入パターン
 - 2)必須範囲の導入パターン
 - 3)商品開発パターン
 - 4)アセット確立パターン
 - 5)プロダクト部品パターン
 - 6)組立てラインパターン
 - 7)監視パターン
 - 8)プロダクト生産パターン
 - 9)ワールドスタートパターン
 - 10)作動パターン
 - 11)プロセスパターン
 - 12)統合運営パターン
 - 13)その他のパターン
- 確認テスト

6. プロダクトラインの導入に向けて

- 1)プロダクトラインアセスメント
- 2)SWPLによる再利用開発
- 3)SWPLの効果
- 4)SWPL導入のポイント
- 5)SWPLへの道のり

7. プロダクトラインの適用事例

- 1)海外事例
- 2)国内事例

到達目標

- セキュリティの基本的な概念を理解する
- セキュリティリスクとその対応方法に関する考え方を理解する
- エンジニアリング内でのセキュリティリスクへの対応方法とそこで活用する技法を理解する

内 容

1. はじめに:目的と目標

2. セキュリティとは

セキュリティに関する現状やセキュリティエンジニアリングの概要を説明します。

- 1)被害事例
- 2)セキュリティの定義
- 3)セキュリティに対する取り組み

3. セキュリティ対策の基本

セキュリティ対策を理解する上で必要になる概念について解説します。

- 1)セキュリティエンジニアリング、セキュリティマネジメント
- 2)セキュリティリスク
- 3)リスクマネジメントプロセス
- 4)セキュリティテスト
- 5)セキュリティバイデザイン

4. セキュリティエンジニアリング

開発内の各フェーズでセキュリティに関してどのような活動を行うのかを解説します。また、活動時に使用する技法についても紹介します。

- 1)全体像
- 2)要求領域での活動
- 3)設計領域での活動
- 4)実装領域での活動
- 5)テスト領域での活動
- 6)運用開始後の対応

5. 安全エンジニアリングとの関係

セキュリティエンジニアリングと並行して実施される可能性の高い、安全エンジニアリングとの関係性について解説します

- 1)安全エンジニアリングとの共通性
- 2)並行して実施するには

6. おわりに

セキュリティエンジニアリング入門講座

近年、システムが社会活動や個人の生活に深く関わるようになって行くに従い、業務系、組込系を問わず、セキュリティへの関心が高まっています。システムからの個人情報流出といった具体的な被害を耳にしたり、システム開発者であれば、脆弱性発見のニュースを目にすることも多くなっていると思います。また、こうした状況をうけ、セキュアなシステムを構築するためのセキュリティ規格・法規制も充実しています(ISO/IEC15408:Common Criteria, ITセキュリティ評価基準)、IEC62443:産業制御システムセキュリティ、ISO/SAE21434:2021:車両サイバーセキュリティエンジニアリング、ISO/IEC 27001:2022 情報セキュリティ管理システムなど)。

製品開発/システム開発においても、こうしたセキュリティに対応していかなければなりません、個別の脆弱性に関する話や、対応策に関する話はあるものの、エンジニアリングの中でセキュリティをどのように扱っていくのかについて触れているものはあまりありません。また、セキュリティと同じように安全(機能安全)についても対応をしなければいけません、安全に関するエンジニアリングとセキュリティに関するエンジニアリングの両方を視野に入れて説明されているものも少ない状況です。

本講座では、「セキュリティエンジニアリング」として、製品開発/システム開発の中でどのようにセキュリティの観点に対応していくのかについて解説します。具体的には、セキュリティ対策における重要なアプローチである、セキュリティバイデザインに基づき、開発ライフサイクル(要求獲得~仕様定義~アーキテクチャ設計~詳細設計~構築~テスト)内でセキュリティリスクをどのように扱っていくのかについて取り上げていきます。また、安全に関するエンジニアリングの活動とセキュリティエンジニアリングの共通性や、それらを並行して実施していくための注意点について解説します。

これからセキュリティ対策を強化して行きたいが、開発の中で具体的にどのように取り組んでいけば良いか悩んでいる方や、そもそもセキュリティって何?という方に、本講座がお役に立つはずです。

日 数

1日

時 間

6時間(10:00~17:00)

受講料(税込)

550,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※オンサイト・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対象者

セキュリティエンジニアリングに携わっている人、今後携わる予定の人(業務系、組込系のどちらの方でも受講対象となる講座です。)

前 提 条 件

システム開発(製品開発)の経験者(ハードウェア開発、ソフトウェア開発は問いません)

オブジェクト指向オリエンテーション

豆蔵トップエンジニアがオブジェクト指向技術のエッセンスを噛み砕いてわかりやすく入門者に伝授いたします。なぜオブジェクト指向が必要とされるようになったのか、その有効性、UMLによる分析や設計と開発プロセスなどの基礎知識が学べます。モデリングとコミュニケーションのためのツールという観点からオブジェクト指向とUMLと反復型プロセスの基本を伝えます。

オブジェクト指向に入門したいというすべての方に開かれた講座です。技術的な要素とビジネス的な要素、入門的な要素と実際の開発イメージとをバランスよく配置し、基礎から実際の開発の全体像まで総合的に理解していただけるオリエンテーション(方向付け)コースになっています。これからオブジェクト技術の導入を図りたい、入門したいという組織の開発エンジニアにも、そのリーダー・管理者にも受講していただく価値がある講座です。日常生活に結び付けて「オブジェクト」概念を理解し、イメージを広げて説明能力を高めていただくための楽しい演習をいくつか実施します。

日 数

1日

時 間

6時間(10:00~17:00)

受 講 料 (税 込)

550,000円

- ※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。
- ※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- 一般的なソフトウェア開発経験者で、今後オブジェクト指向開発を目指す方
- プログラミングやソフトウェア開発に関する初歩的な知識を有する新人や管理者で、オブジェクト指向技術とは何かを知りたい方

前 提 条 件

特になし
(一般のソフトウェア開発に関する基本的な知識ないしは経験があると理解が容易になります)。

訓 達 目 標

- オブジェクト指向開発の見取り図(全体イメージ)を頭に入れることができます。
- オブジェクト指向技術のソフトウェア開発における有用性が理解できます。
- オブジェクト指向の基本的な概念や標準的な用語を説明できるようになります。
- モデリング言語 UML の特徴が理解できます。
- 反復型開発プロセスの特徴が理解できます。

UML 入門講座

オブジェクト指向におけるモデル表記法であるUML(Unified Modeling Language)は、1997年11月のバージョン1.1が公開されてから現在まで着実に進歩/普及を続け、今や世界的なデファクト・スタンダードの地位をしっかりと確立しています。MDA(Model Driven Architecture)やコンポーネント・ベース開発などへの対応/適用も図られ、ビジネス系/組込み系など分野の違いにかかわらず、モデル中心型の開発スタイルを目指す開発者にとって、UMLはほぼ必須のスキルになってきたと言って良いでしょう。

UMLに関しては、さまざまなホームページや書籍などで情報を収集することができますが、そういった情報源をもとに独学した人の中には、曖昧な解釈や間違った思い込みをしまっている人もチラホラ見かけます。また、初めてUMLを勉強する人にとっては、ボリュームが大きく、なかなか体系的かつ効率的に学習することができません。初期(バージョン1.*)の頃にUMLを習得された方にとっても、UML2.*で新たに追加された図や解釈が変わった表記要素などがあるので、全体的に一回はきちんと勉強し直す必要があります。

本講座では、UMLで定義されているおもな図/要素の表記法について、1日で体系的かつ効率的に習得していきます。特に良く使われる図のいくつかについては、簡単な演習も交えて体験的に理解を深めることができるようになっています。

UMLモデルを活用するオブジェクト指向開発に携わるすべての開発者にオススメの講座です。

なお、本講座には「初學者のためのUML入門講座」というバリエーションもご用意しています。これは、C言語を主に利用するエンジニアなどオブジェクト指向言語を用いない方を対象としたもので、インターフェイスやポリモーフィズムといった高度な概念や複雑なステートマシン表記などを省き、代わりにダイアグラムの全体像や連携を理解しやすとした、初めて学ぶ方でも安心して取り組める内容です。特に、C言語で開発するエンジニア、モデルベースで要件定義をしたい方に適した講座となっています。

日 数

1日

時 間

7時間(9:00~17:00)、初學者向け7.5時間(9:00~17:30)

受 講 料 (税 込)

550,000円

- ※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。
- ※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- はじめてUMLを学んでみようという方
- オブジェクト指向開発に携わる全ての方
- オブジェクト指向を使用していない場合でも例えば以下の職種作成者 / 要求分析者 / 設計者

前 提 条 件

- オブジェクト指向の基本的な概念を習得している方
(「オブジェクト指向オリエンテーション」講座受講済み相当の知識がある方)

内 容

形式: 講義+演習(個人+グループ)

1. コースの目的

2. なぜオブジェクト指向か

現在のソフトウェア開発においてなぜオブジェクト指向技術が必要とされるかについて学びます。

- 1) 現在のソフトウェア開発の課題
- 2) オブジェクト指向への道のり
- 3) オブジェクト指向の利点

3. オブジェクト指向の概念

オブジェクト指向のもとになる考え方・用語と概念を基本から学びます。

- 1) オブジェクトとは何か?(オブジェクトのもつべき性質)
- 2) オブジェクト指向の構成要素(クラス、カプセル化、継承、ポリモーフィズム)

4. モデリングとUML

要求・分析・設計のためのモデリング言語UMLの有用性とUMLダイアグラムの概要について学びます。

- 1) モデリングとオブジェクト指向分析・設計・実装
- 2) 統一モデリング言語UMLの概要

5. オブジェクト指向開発

要求とスクのコントロールを重視した反復型開発プロセスの特徴、従来型開発との相違について学びます。

- 1) 開発モデル(ウォーターフォールモデル、スパイラルモデル、オブジェクト指向モデルの概要)
- 2) 開発プロセス(統一プロセスとアジャイルプロセスの概要)

6. Object-Oriented Orientation セミナーまとめ

7. 付録: 参考文献・補足資料

オブジェクト指向開発に付随して知っておくべき関連技術の概要について学びます

- 1) プログラミング言語、永続化、分散、コンポーネントとフレームワーク、Webサービス、ビジネスモデリング他

※ いくつかの章で、内容の概念理解を深めるための演習を行います。

訓 達 目 標

- UMLとは何か、採用のメリットを人に説明できるようになる。
- UMLの学習した範囲内で記述されたモデルであれば(ie 大方のモデルが)理解できるようになる。
- 学習したUMLの図のいくつかを現場で使い始められるようになる。

内 容

【目次】

1. はじめに

- 1) 本講座の目的/目標
- 2) 講義進行の目安

2. UMLの概要

- 1) UMLとは
- 2) UMLが提供するもの
- 3) UML普及の理由/効果
- 4) UML利用事例

3. オブジェクト指向の基本概念

- 1) オブジェクトとは何か
- 2) オブジェクトのもつ4つの特徴
- 3) クラス(Class)
- 4) インスタンス(Instance)
- 5) カプセル化(Encapsulation)
- 6) 汎化関係(継承)(generalization (inheritance))
- 7) 継承(Inheritance)
- 8) ポリモーフィズム(多相性)(Polymorphism)
- 9) インターフェイス(Interface)

[4. ~ 16. UMLのダイアグラム]

4. ユースケース図

5. クラス図

6. オブジェクト図

7. シーケンス図

8. 相互作用概念図

9. コミュニケーション図

10. タイミング図

11. ステートマシン図

12. アクティビティ図

13. コンポーネント図

14. コンポジット構造図

15. パッケージ図

16. 配置図

17. おわりに

5章の詳細内容(他の章もほぼ同様の構成)

1)基本表記要素

クラス、汎化/継承、関連、集約...

2)その他重要な表記要素

クラス定義に関する補足、

汎化/継承に関する補足、

関連に関する補足...

オブジェクト指向プログラミング C++ 演習講座

組込系ソフトウェア開発の世界においても、マイコンの高性能化、メモリの低価格化を受け、オブジェクト指向開発が採用されることも多くなってきています。しかし UML による設計や C 言語による実装は出来ても、オブジェクト指向言語による開発は出来ない方が多いのではないのでしょうか？

本講座は、代表的なオブジェクト指向言語である C++言語を対象に、プログラミング言語研修であるような文法中心でなく、C 言語にはない C++言語の実装方法、UML による設計内容の実装方法といった実践的な内容を演習中心に行い、オブジェクト指向開発での C++言語の利用法を習得する講座になっています。

日 数

3 日
4 日

時 間

3 日 7 時間 / 日 (9:00~17:00)
4 日 6 時間 / 日 (10:00~17:00)

受講料 (税込)

3 日:1,650,000 円、4 日:2,200,000 円
※オンラインでの開催となります。
※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

オブジェクト指向開発の実装を C++言語で行う方

前 提 条 件

- UML の基本的な表記法を習得している方
- C 言語によるソフトウェア開発の経験者(2 年以上が望ましい)

到達 目 標

- UML で記述されたモデルに従って、C++言語にてプログラミングが行える。
- C++言語と C 言語との違いを第三者に説明できる。

UML / C・C++ マッピング講座

設計者としてクラス図を書いているけれど、自分が書いていることが実装にどう反映されるのか分からない」「実装者として実装が終わった後、納品に必要なので、言われたところだけクラス図を書いています」という方はいませんか。

UML もオブジェクト指向言語も、同じ「オブジェクト指向」という考えに基づいており、両者には対応付けられる部分があります。この対応付けをマッピングと呼びます。マッピングを正しく理解することで、設計者は設計意図をより正確に実装者に伝えられるようになり、実装者は設計意図を理解した上でコードを書けるようになります。

特に「モデルなんか知らなくても実装はできる」と考えている方には、UML と実装のマッピングを学習することをオススメします。クラス図やステートマシン図やシーケンス図を実装と対応付けて理解することで、ソフトウェアの全体構造を把握しながら実装できるようになります。

本講座の特徴は以下の通りです：

- UML の記述内容が実装にどう結びつくのかを短時間で習得できる
- クラス図やシーケンス図を実装と照らし合わせて理解できるため、ソフトウェアの構造と振る舞いの理解が深まる
- 設計と実装を結びつけ、変更影響の把握や保守性の向上に役立つ

本講座は C 言語コース と C++コース の 2 種類をご用意しています。

- C++コース:クラス、継承、テンプレートなどを UML と対応付け、オブジェクト指向設計を正しくコードに落とし込みます。
- C コース:非オブジェクト指向言語である C においても、関数やデータ構造を UML モデルと対応付け、設計情報を活かした開発手法を体験できます。

日 数

1 日
2 日

時 間

1 日 6 時間 (10:00~17:00)
2 日 6 時間 / 日 (10:00~17:00)

受講料 (税込)

1 日:550,000 円、2 日:1,100,000 円
※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
※オンライン・オンライン (Zoom 等のオンライン会議システムを利用して実施可能)ともに同料金です。
※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- UML を使った開発を始めたい方、UML の実装に対する効果を知りたい方

前 提 条 件

- オブジェクト指向プログラミング言語(C 言語、C++)の基本的な文法の知識(制御構文、変数、配列、関数、クラス定義)。
- 何らかのプログラミングの経験があると理解が容易になります(必須ではありません)。
- UML の表記法をあらかじめ知っているとう理解が容易になります(必須ではありません)。

内 容

形式:講義+演習(個人)

【第 1 日目】

1. はじめに
講義の目的・目標を説明し、受講者が理解して欲しいこと、また進行の目安について説明します。
2. OOP(オブジェクト指向プログラミング)概要
構造化とオブジェクト指向の違いをモデリングと言語の観点で解説します。
3. 基本データ型と関数定義
C++言語で利用できる型の解説をします。演習環境(Visual Studio)の環境に慣れる演習を行います。
4. 自動変数と動的オブジェクト確保
C 言語と比較しながら C++言語の自動変数と動的オブジェクト確保の解説をします。[演習]
5. クラスの定義
UML 表現のクラスを C++言語表現にする方法とインスタンス生成を解説します。[演習]
6. カプセル化
カプセル化を C++言語で表現する方法とアクセス制御を解説します。[演習]

【第 2 日目】

7. オブジェクトの生成と消滅
コンストラクタ、デストラクタ、初期化子を解説します。[演習]
8. オブジェクトの生存期間
3種類のライフサイクル、クラス変数とクラスメンソバ、this ポインタを解説します。[演習]
9. 関連とメッセージ送信
UML 表現(クラス図、シーケンス図)のマッピング方法を解説します。[演習]
10. 集約、コンポジション、依存
UML 表現の集約・コンポジション・依存のマッピング方法を解説します。[演習]
11. その他の関連
UML 表現の多重度の関連、双方向の関連、関連クラスのマッピング方法を解説します。[演習]

【第 3 日目】

12. 継承
UML 表現の継承関係を C++言語の継承へのマッピング方法を解説します。コンストラクタ・デストラクタの呼び出し順、基底クラスの参照、自動ポインタも解説します。[演習]
13. 関数オーバーライド、仮想関数
UML 表現の多相性のマッピング方法、関数のオーバーライド、仮想関数を解説します。[演習]
14. 純粋仮想関数
UML 表現のインタフェース、抽象クラスのマッピング方法、純粋仮想関数を解説します。[演習]
15. インタフェースクラス
インタフェースの利点、UML の実現関係とインタフェースクラスを解説します。
16. 委譲
UML 表現の委譲のマッピング方法、継承から委譲への変形を解説します。[演習]

【第 4 日目】

17. オブジェクトの状態遷移
UML 表現のステートマシン図からのマッピング方法を解説します。[演習]
18. 参照
参照型の意味とその利用方法を解説します。[演習]
19. コピーコンストラクタ、代入演算子
コピーコンストラクタ、代入演算子の意味とその利用方法を解説します。[演習]
20. ストリーム、const
ストリーム、const の意味とその利用方法を解説します。[演習]
21. コレクションクラス
コレクションクラスの意味とその利用方法を解説します。[演習]
22. テンプレート、STL、スマートポインタ
テンプレート概念、STL の概要、auto_ptr の概要を解説します。[演習]
23. 名前空間
名前空間の概要、利用方法を解説します。
24. パッケージの実装
C++のパッケージ実装の概要、UML 標記との関係を解説します。[演習]
25. デザインパターン
デザインパターンの概念、Observer/Proxy パターンの実装例を解説します。

到達 目 標

- UML のクラス図、シーケンス図、ステートマシン図を基にプログラムが書ける。
- モデリングツールを使って、上記のダイアグラムが描ける。

内 容

形式:講義+演習(個人)

1.はじめに

- 1.1 目的と目標
- 1.2 前提条件
- 1.3 講義進行の目安

2.モデルベース開発

- 2.1 モデルベース開発とは
- 2.2 なぜモデルベース開発が必要なのか
- 2.3 モデルを活用する場面
- 2.4 モデルベース開発におけるエンジニア像
- 2.5 モデルベース開発で使う技術

3.モジュール単体の定義

- 3.1 ソフトウェアとモジュールの関係
- 3.2 モジュール単体の静的構造
- 3.3 モジュール単体の動的構造
- 3.4 単体のモジュールを動かす

4.モジュールの連携

- 4.1 モジュール間の関係
- 4.2 モジュールの連携
- 4.3 連携するモジュールを動かす

5.その他コード化時の注意点

- 5.1 コード化時の注意点

UML による設計可視化講座



仕様変更や不具合対応をする時、どこを修正すればよいか、修正したらどのあたりに影響が出るのかといったことが、なかなか分らなくて困ったことはないでしょうか。また、ソフトウェアの作りを見直すにも、ソースコードの規模が大きくなりすぎて、ソースコードが読み切れない、ソフトウェアの全体像が把握出来ないといったことはないでしょうか。その原因は設計(ソフトウェアの構造)が見えなくなっていることにあります。本講座は、そうした設計情報が無くなってしまっている方、そうならないよう、設計情報を残しながら開発を進めていきたいと考えている方にお勧めの講座です。

本講座では、UML とソースコードの対応関係について徹底的に学びます。UML(クラス図、ステートマシン図、シーケンス図)で描かれた内容をプログラムに変換するフォワードエンジニアリングだけでなく、ソースコードに書かれた情報から UML のダイアグラム(クラス図、ステートマシン図)を作成し、設計を可視化するリバースエンジニアリングの双方を学ぶことで、UML による記述とプログラムの対応関係がイメージできるようになります。このイメージができると、プログラムを見なくてもソフトウェアの設計を検討することができるようになります。また、本講座では設計原則についても取り上げているので、UML 上で、ソフトウェア設計を改善していく方法についても習得することができます。

設計の可視化は、ソフトウェアの全体像の理解、不具合や仕様変更の影響範囲の把握、設計の悪いところの発見・改善に対して効果あります。これらのことで困っている方、大量のソースコードのメンテナンスで悩んでいるという方に、最適な講座です。

また本講座は、UML の基本的な読み書きから取り上げていますので、UML の知識が無い方も安心して受講して下さい。

日 教

- 1 日
- 2 日

時 間

- 1 日 6 時間(10:00~17:00)
- 2 日 6 時間 / 日(10:00~17:00)

受 講 料 (税 込)

1 日:550,000 円、2 日:1,100,000 円

- ※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
- ※オンライン(Zoom 等のオンライン会議システム)を利用しても実施可能)とも同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- UML のソフトウェア設計への適用を考えている方
- ソフトウェアの構造が把握出来なくて困っている方
- ソフトウェアの設計の見直しを考えている方

前 提 条 件

- C 言語によるソフトウェア開発の経験があること
- UML についての知識は前提としていません

UML モデリング演習講座



— オブジェクト指向の開発(分析・設計)に必要なモデリングスキルを演習を通し習得する —

オブジェクト指向は、分析にせよ設計にせよ、オブジェクトを使って対象を分解・構成していくところに特徴があります。この「分解・構成」に必要なものがモデリングスキルであり、モデリングスキルはオブジェクト指向開発にとって最も重要な基盤的なスキルといえます。

これまでもモデリングをとりあげたコースは、弊社の教育コースも含め、数多くありました。ただ従来のコースでは、モデリングは UML や開発プロセスや分析といったものと一緒にとりあげられており、モデリングスキルを学ぶには必ずしも十分な構成ではありませんでした。

本講座はこれに対し、モデリングスキルが十分学べるよう、モデリングだけにフォーカスした構成になっています。弊社は従来より無償の学びの場として「モデリング道場」なるものを開催してまいりましたが、その教育コース版といえます。

本講座は演習形式の講座です。演習問題は今回新たに弊社のこれまでの教育経験・コンサルティング経験を基に精選したもので、2 日間コースでは全 10 題を扱い、実践に必要な各モデリングスキルを一通り習得できる構成になっています。1 日間コースも用意しており、その場合は主要な演習問題を中心に取り上げ、効率的に基礎を習得できるよう工夫しています。

初心者はもちろん、ある程度モデリング経験がある中級者にとっても、有意義な講座になっています。是非、一緒に楽しみながら、モデリングスキルを磨いてみませんか！

日 教

- 1 日
- 2 日

時 間

- 1 日 6 時間(10:00~17:00)
- 2 日 6 時間 / 日(10:00~17:00)

受 講 料 (税 込)

1 日:550,000 円、2 日:1,100,000 円

- ※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
- ※オンライン(Zoom 等のオンライン会議システム)を利用して実施可能)とも同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

開発者(設計者・実装者)、要求分析者、プロジェクト管理者

前 提 条 件

- オブジェクト指向の基本概念(クラス、インスタンス、継承)を理解していること。
- UML の基本要素(クラス図、シーケンス図、ステートマシン図)が使えること。

例 達 目 標

- モデルによって、伝達したいことを表現できるようになる。
- オブジェクト指向モデリングの要領・勘所を習得し、通常のレベルの問題であれば適切にモデリングできるようになる。

例 達 目 標

- UML の図を書けるようになる
クラス図、ステートマシン図、シーケンス図、アクティビティ図、タイミング図、コンポーネント図
- UML の図をプログラムに変換できるようになる
- プログラムから UML の図を作成できるようになる
- UML を使ってメカニズムを表現できるようになる
- 設計原則を人に説明できるようになる
- クラス図上で設計の検討が出来るようになる
注) 上流工程に相当する内容は、このコースに含まれません

内 容

形式: 講義 + 演習(個人)

1. はじめに

- 1) 目的と目標
- 2) 前提条件
- 3) 講義進行の目安

2. ソフトウェア設計の改善とは

- 1) ソフトウェアの構造とは
- 2) ソフトウェア設計の進化
- 3) リバースエンジニアリング
- 4) リバースエンジニアリングとリエンジニアリングの関係

3. モジュール単体の定義

- 1) ソフトウェアとモジュールの関係
- 2) モジュール単体の静的構造
- 3) モジュール単体の動的構造
- 4) 単体のモジュールを動かす

4. モジュールの連携

- 1) モジュール間の関係
- 2) モジュールの連携
- 3) 連携するモジュールを動かす

5. その他コード化時の注意点

- 1) コード化時の注意点

6. リバースエンジニアリング

- 1) リバースエンジニアリングとは
- 2) 全体手順
- 3) 構造のリバース
- 4) 動きのリバース

7. 良い設計とは

- 1) 悪い設計
- 2) 良い設計
- 3) 良い設計にするためには
- 4) トレードオフ

内 容

1. はじめに

- 1) 本講座の目的
- 2) 本講座を受けての達成目標
- 3) 前提条件
- 4) 講義進行の目安

2. モデリングとは？

- 1) モデルとは
- 2) モデルの例
- 3) モデルの特徴
- 4) モデルに対する考え方
- 5) モデルの効果

3. オブジェクト指向とモデリング

- 1) オブジェクト指向開発のポイント
- 2) オブジェクト指向システムにおけるモデリング
- 3) 組込み OS でのモデルの考え方
- 4) モデルを活用する場面
- 5) モデルベース開発におけるエンジニア像
- 6) モデルベース開発で使う技術

4. オブジェクト指向モデリングの進め方

5. 静的構造のモデリング

6. 動的構造のモデリング

7. 相互作用の可視化

- 1) 相互作用とは
- 2) 作成の基本手順
- 3) 相互作用図の注意点

8. モデリングの共通留意事項

※ 講義の中で演習問題(10 問)も行います。

UML モデリング添削講座

本講座は演習問題を通して UML モデリングスキルの向上を図る講座です。
UML は表記法を覚えるだけでなく、モデリングのために必要な思考法を身につけて初めて役に立つ技術です。そのような思考法を身につけるためには、多くのタイプの問題にぶつかり、考え方の誤りを指摘されることが不可欠です。しかし、通常の業務の中では多くのタイプの問題に当たることも、誤りを指摘されることもそうありません。そのため、モデリングスキルを身につけるには非常に多くの時間を必要とします。

そうした状況に対して、この講座では、モデリングスキル向上に必要な理想の環境を提供します。つまり、徹底的に多くの問題を解き、モデルの誤りを指摘していきます。

出題される問題は、弊社の培ってきた経験から導かれた、「モデリングスキルの向上に必要なポイント」に基づいて作られています。このポイントを網羅するように作られた問題は 50 問以上に渡り、その質の高さ、バラエティの豊富さは他に類を見ないものとなっています。(出題されるのはその中から選ばれた問題です)

また、この講座は添削型であることも特徴です。
出題→回答→解答配布で終わってしまうモデリング講座が多い中、本講座は受講作成したモデルひとつひとつに対して、添削を行います。この添削により、自身のモデルの点、悪い点が把握できるため、独力でモデリングの学習をした場合よりも効果的に、短時間でモデリングスキルの向上を図ることができます。また、添削は通信添削の形のため、場所、時間を問わずに受講することができますので、業務後や週末の空いている間を利用して、効果的にモデリングスキルを身につけていくことができます。

問題は難易度に応じてレベル 1～レベル 2 まで用意されており、レベル 1 は業務系、込み系に共通の問題、レベル 2 は業務系、組込み系それぞれに対応した問題を用意しています。用意された問題を全てモデリングした頃には、あなたのモデリングスキルは確実にアップしています。



生の良かつると時組して実際に

レベル	対象範囲	内容
レベル 1	問題領域に依存しない問題	UML の表記法の理解を問う問題。
レベル 2	業務系/組込み系を選択	間違いやすいポイントを含む問題。ダイアグラムを組み合わせる問題。

問題のレベル分け

モデリング技術はソフトウェア技術者にとって非常に重要な基盤技術です。UML を覚えた後の次のステップとしてだけでなく、さらにも上のモデリングスキルを身に付けたい方、また、これから業務でモデリングをする方、業務とは関係なく技術者としてのステップアップを目指している方も、是非、この講座を受講してみてください。

各レベルの概要

レベル	日数	添削サイクル	1 サイクル当りの出題数	合計出題数
レベル 1	1.5 ヶ月～	3 サイクル	3 問	9 問
レベル 2	3.0 ヶ月～	6 サイクル	2 問	12 問

※ レベル 1 から受講してください

受講料 (税込)

サイクル数や出題数は、ご相談の上、調整可能です。お問い合わせください。

対象者

UML モデリングのスキル向上を図りたい方

前提条件

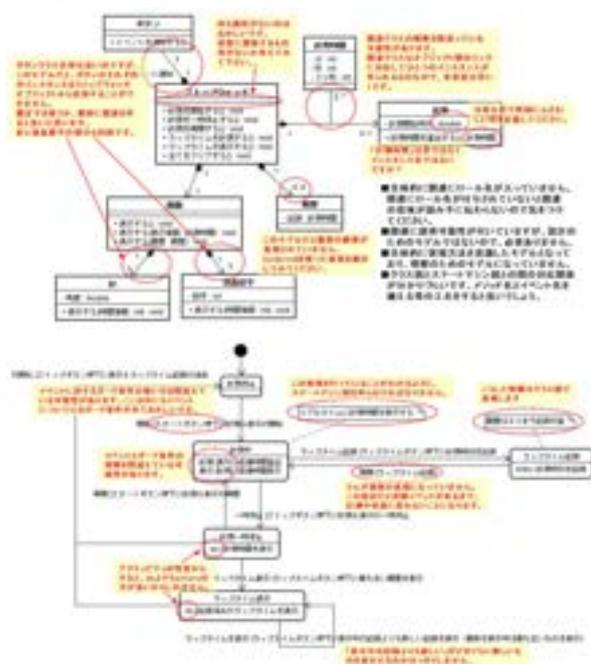
- オブジェクト指向の基本的な概念を習得している方
- UML の基本的な表記法を習得している方
 - UML 1, OCAP Fundamental 取得相当の知識が必要です

到達目標

- モデル要素を正確に適用した UML のダイアグラムが描けるようになる
- 複数のダイアグラムを組合わせても、矛盾が生じないモデルを作れるようになる
- 実開発における複雑な事象を読み解きモデル化できるようにする

内容

【添削例】
ストップウォッチ(ラップタイムを4つ保存可能)をモデリングしてくださいという問題に対する回答と添削例(2 相当)。



UseCase 入門講座

本講座は UseCase の実践的な利用方法に関する講座です。特に、組込み系での UseCase の適用と活用について取り上げていきます。

近頃、UML の普及とともに様々な開発で UseCase が使われるようになってきました。そうした中、少なくなってきたはいるものの、UseCase がどのような効果を持つのか理解できていなかったり、効果の出ない作り方/使い方をしている場合に遭遇することがあります。この講座では、そのような状況を解決するために、UseCase を使うことによる効果、正しい作り方、使い方を説明していきます。また、UseCase 単独での効果だけでなく、開発全体を通して、UseCase を他のプロセス内でのように活用するのについても触れています。

また、この講座は組込み系での UseCase の適用方法について触れていることも特徴です。組込み系の開発でも UseCase が使われるようになってきていますが、組込み系特有の割込みやイベントといった概念と UseCase の関係や、部分システムに対する UseCase の適用について取り上げている書籍や記事はそれほど多くありません。本講座はそうした部分も取り上げ、上記のような UseCase の効果や作り方、使い方、他プロセスとの活用に対して、それぞれの章で、組込み系での場合について触れています。また、その内容は現場の経験に基づくものであり、実践的なものになっています。

この講座ではテキストに対応した演習問題を用意しており、演習問題を通して自ら手を動かすことで、理解を深めることができます。ようになっています。また、この演習問題は組込み系を題材としていますので、現実に即した経験をするすることができます。

組込みを対象とした UseCase の講座は、まだ非常に珍しい存在です。UseCase が組込み系で役に立つのかを知りたい方、UseCase の開発への適用を考えている方、UseCase を使っているが効果を感じられない方、この期に、ぜひともこの講座を受講してみてください。

日数

1 日

時間

6 時間 (10:00～17:00)

受講料 (税込)

550,000 円

- ※ 理想受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
- ※ オンサイト・オンライン (Zoom 等のオンライン会議システムを利用して実施可能) ともに同料金です。
- ※ 受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対象者

- UseCase の組込みソフトウェアでの適用方法を知りたい方
- UseCase を作るようになりたい方
- UseCase 作成のスキルを向上させたい方

前提条件

- なんらかのソフトウェア開発の経験があること
- UML、オブジェクト指向の知識は必要ありません

到達目標

- UseCase を作る意義を人に説明できるようになる
- シナリオ分析の意義を人に説明できるようになる
- UseCase の後工程での活用方法を人に説明できるようになる
- UseCase を読み書きできるようになる (UseCase 図、UseCase 記述、シナリオリスト、シナリオ詳細記述、部分システムに対する UseCase)

内容

- はじめに
- UseCase とは
- エンジニアリングプロセスにおける UseCase の位置づけ
 - 1) UseCase の位置付け
 - 2) システム要求分析の位置付け
 - 3) システム要求分析とは
- UseCase の表記法
 - 1) UseCase 図
 - 2) UseCase 記述
- UseCase の作成
 - 1) UseCase 作成の手順
 - 2) UseCase 作成の注意点
- シナリオ分析
 - 1) シナリオ分析とは
 - 2) シナリオの表記法
 - 3) シナリオ分析の進め方
- UseCase とシナリオ分析の効果
 - 1) UseCase の効果
 - 2) シナリオ分析の効果
- UseCase と他プロセス・作業成果物との関係
 - 1) 要求との関係
 - 2) 要件/仕様との関係
 - 3) 分析における UseCase の利用
 - 4) 設計における UseCase の利用
 - 5) テストにおける UseCase の利用
 - 6) プロジェクトマネジメントと UseCase の関係
- サブシステムが対象の場合の UseCase
 - 1) サブシステムを対象とした UseCase とは
 - 2) 開発での使い方
 - 3) アクタの特定
 - 4) 相互作用の粒度
 - 5) その他留意点



オブジェクト指向分析設計講座

— 組込みシステム開発向け —

本講座は、オブジェクト指向・UMLを使用したシステム開発を、要求定義～要求分析～設計という工程を追いながら学んでいきます。組込システム開発へのオブジェクト指向適用に関しては、まだまだ資料も実践的なセミナーも少ないのが現状ですが、本コースを受講することで、組込システム開発へのオブジェクト指向適用の意義を知り、またシステム開発にオブジェクト指向を適用するための基本的な部分を習得することができます。

本講座の特徴としては、演習を中心とした実践的な講座であることが挙げられます。組込系のシステムである自動販売機を演習問題とすることで、組込特有のハードウェアとその制御をモデルとしてどのように扱うのかといった課題を演習を通して実際に解決しながら習得していくことができます。さらに演習で受講生の作成したモデルは、講師がその場でレビューし、モデル品質向上のためのアドバイスを致します。このレビューを通してモデルの良い点、悪い点を具体的に知ることができ、より良いモデリングを習得していくことができます。また、講師は組込システムのコンサルティングを経験していますので、実際の経験に基づいた実践的なアドバイスが可能です。

この演習問題については、お客様のご要望に合わせて変更することもできます。現在開発しているシステムを演習問題とし、よりオブジェクト指向/UMLに対する理解を深めるといったカスタマイズを行うことも可能です。

本講座はシステム開発全体像とオブジェクト指向分析設計の対応関係が明確になっているため、ソフトウェア開発経験のある技術者に対するオブジェクト指向の導入を検討している場合や、オブジェクト指向適用の現状を知りたい、開発チームに対するオブジェクト指向の導入教育を考えている、といった場合に適したコースです。また、すでにオブジェクト指向/UMLを開発に導入している方にも、モデリングスキル向上のための講座として有効です。

なお、本講座は短期集中で効率的に学びたい方には2日間コース、演習時間をじっくり確保して深く学びたい方には3日間コースをご用意しています。目的やご都合に合わせて選択いただけます。

日数

3日

時間

6時間/日(10:00~17:00)

受講料(税込)

1,650,000円

- ※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。
- ※オンサイト・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対象者

- 組込みシステム開発におけるオブジェクト指向の適用方法を知りたい方
- UMLの組込みシステム開発向けの使い方を知りたい方

前提条件

- オブジェクト指向の基礎用語(クラス、インスタンス、継承、カプセル化など)を理解している。
- UML表記法の基礎を理解している。
- Java、C、C++などのプログラミング言語で簡単なプログラムを作成したことがあれば、理解は容易になります(必須ではありません)。
- 何らかのソフトウェア開発経験があれば、より内容を理解しやすくなります



デザインパターン入門

オブジェクト指向設計をこれから始めよう、もっと良いオブジェクト指向設計ができるようになりたいという時に学ぶと効果的なのが、です。

デザインパターンはオブジェクト指向設計のためのベストプラクティス集であり、これを学ぶことで、オブジェクト指向設計と構造化設計の違いはどこなのか、オブジェクト指向設計の本当の効果はどこにあるのか、といったオブジェクト指向設計のエッセンスを学ぶことができます。

一方、既にデザインパターンを適用しているという現場でよく見られるのが、パターンを適用することが目的となってしまうという問題です。そうした現場では、デザインパターンが設計課題の解決になっていないばかりか、ソフトウェア構造の複雑化を招いてしまい、逆効果となっていることもあります。

そうした現場での状況も踏まえ、本講座ではデザインパターンの紹介だけでなく、分析モデルに対してデザインパターンを適用し、設計課題を解決する設計モデルをつくるという実践的なデザインパターンの使用方法を取り上げています。

コース名	特徴	所要日数
概要編	・デザインパターンの意義とGoF本『デザインパターン』で紹介されている23種類のパターンについての解説を行います。	1日
適用編	・具体的な題材を使った演習をベースに、各種のパターンを適用し、課題解決を行っていく過程を解説していきます。	1日
コンパクト版	・概要編と適用編をコンパクトにまとめ、1日間で効率的に学習する講座です。	1日

これからオブジェクト指向設計を始める方にも、既にオブジェクト指向で設計している方にも有用な講座となっていますので、是非受講してみてください。

日数

- 概要編: 1日
- 適用編: 1日
- コンパクト編: 1日

時間

- 概要編: 6時間(10:00~17:00)
- 適用編: 6時間(10:00~17:00)
- コンパクト編: 6.5時間(10:00~17:30)

受講料(税込)

- 概要編: 550,000円
- 適用編: 550,000円
- コンパクト版: 550,000円

- ※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。
- ※オンサイト・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対象者

- 【概要編】
 - デザインパターンとは何かを知りたい方

【適用編】

到達目標

- 組込みソフトウェア開発に対するオブジェクト指向の適用方法が分かるようになる。
- オブジェクト指向を利用した場合の開発手順(要求定義、要求分析、設計)が理解できるようになる。
- UMLを用いた簡単なモデリングができるようになる。

内容

【第1日目】
形式: 講義+演習(個人+グループ)

- はじめに
- プロセス概要
- 要求定義
 - 1) 要求定義の目的
 - 2) 要求の獲得からシステム要件作成まで
 - 3) 要求定義の手順
 - 4) ユースケース図の作成
 - 5) ユースケース記述の作成
 - 6) ユースケース作成のポイント
 - 7) ユースケースの効果
 演習: UseCase図の作成、UseCase記述の作成、シナリオ記述の作成
- 要求分析
 - 1) 要求分析の目的
 - 2) 要求分析の手順
 - 3) シナリオ分析
 - 4) オブジェクト指向分析
 - 4.1) オブジェクト指向分析の手順

【第2日目】
形式: 講義+演習(個人+グループ)

- 4.2) 組込系固有の問題
- 4.3) オブジェクト指向分析のポイント
- 4.4) オブジェクト指向分析の効果
- 演習: クラス図の作成、シーケンス図の作成
- 設計
 - 1) オブジェクト指向設計とは
 - 2) 分析と設計の違い
 - 3) 設計手順
 - 3.1) アーキテクチャ/方針の決定
 - 3.2) 対象となる分析モデルの確認
 - 3.3) 対象となる要件の確認
 - 3.4) 分析モデルをベースとした設計
 - 3.5) 設計課題の検討
 - 3.6) 設計手順の計画
 - 3.7) 対象とする上位要件の選択
 - 3.8) 解決先の選択
 - 3.9) 設計モデルの作成
 - 3.10) トレース
 演習: クラス図の作成、シーケンス図の作成、ステートマシン図の作成

6. おわりに

【3日間開催を選択いただいた場合は、以下の内容を追加し、全体を3日間で実施します】

7. 設計詳細
 - 1) 変更可能性のある部分の分離
 - 2) 並行処理の実現
 - 3) メモリ管理
 - 4) ROM/RAMサイズの最適化

- デザインパターンは知っているけど、使い方が分からないという方
- オブジェクト指向設計のスキルを向上させたい方

前提条件

- 【概要編】
 - オブジェクト指向、UMLに関する基本的な知識

- 【適用編】
 - デザインパターンについての基本的な知識(パターン名を聞いて、特徴が分かる程度)

到達目標

- 【概要編】
 - デザインパターンとは何かを人に説明できるようになる
 - 講座で取り上げているデザインパターンについてはその適用場面、効果を説明できるようになる
- 【適用編】
 - 自身の設計にデザインパターンを適用できるようになる

内容

<<概要編>>

- 形式: 講義
1. デザインパターン概観
 - 1) 動機と由来
 - 2) 現状の開発における問題認識
 - 3) 設計ノウハウとは?
 2. デザインパターン入門
 - 1) 要約
 - 2) 基本構成
 - 3) メタモデル
 - 4) パターンの構成と表記法
 - 5) UMLでのデザインパターン表記法
 - 6) パターン・ムーブメント
 - 7) 狭義のデザインパターン
 - 『デザインパターン』本の内容
 - ・生成のための5パターン
 - ・構成のための5パターン
 - ・振舞いのための11パターン

<<適用編>>

- 形式: 講義+演習
- 1 はじめに
 - 4.4 Composite
 - 2 オブジェクト指向設計のエッセンス とデザインパターン超入門
 - 5 デザインパターン振舞編
 - 5.1 振舞パターンとは
 - 5.2 Template Method
 - 5.3 Observer
 - 5.4 State
 - 5.5 Command
 - 5.6 Mediator
 - 3 デザインパターン生成編
 - 6 適用事例
 - 6.1 事例の概要
 - 6.2 要求モデル
 - 6.3 分析指針
 - 6.4 分析モデルの適用例
 - 6.5 設計指針
 - 6.6 設計モデルの適用
 - 6.7 演習問題

オブジェクト指向開発プロセス

— オブジェクト指向開発の概要、活用のポイントについて理解する —

昨今の UML の普及により、オブジェクト指向開発を開発現場に適用するケースが多くなってきました。しかしながら「過去行っていた開発プロセスをそのまま利用」「オブジェクト指向開発プロセスを理解せずに適用」などのために開発現場が混乱し、オブジェクト指向を導入した狙い・目的(再利用による生産性の向上、モデル化による設計の可視化など)を達成できずに頓挫するケースが多々見られるのが実情です。その1つの原因が、オブジェクト指向開発のプロセスを真に理解せずに、開発に着手することにあります。

本コースは、開発プロセスの目的・価値・活用方法を解説すると共に、オブジェクト指向開発の開発現場で多く適用されている RUP(Rational Unified Process)、XP(eXtreme Programming)の内容と適用方法を解説し、オブジェクト指向開発プロセスを素早く理解してもらう教育コースです。

豆蔵では、長年蓄積してきたオブジェクト指向の開発技術及びオブジェクト指向開発プロセスの導入支援、並びに CMM/CMMI の導入支援の成果で、オブジェクト指向開発の普及に取り組んでいます。そこで得た経験、ノウハウを基に作ったのが本コースです。

オブジェクト指向開発とは何かを理解したい方、あるいはさせたい方、これからオブジェクト指向を開発に適用しようとする方と検討されている方に最適です。どうぞ、お試しあれ！

日 数

1 日

時 間

7 時間(9:00~17:00)

受講料 (税込)

550,000 円

※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。

※オンサイト・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

プロジェクトマネージャ/リーダー、アーキテクト、現場開発者、現場管理者、プロセス改善推進者、上位管理者

前 提 条 件

ソフトウェアに関する基礎知識を持っている事

訓 練 目 標

- そもそも開発プロセスは、どのような構成になっているかを理解し、その中でどのような活動があるかを把握し、他のメンバーに簡単に説明ができるようになることを目標としている。
- オブジェクト指向開発プロセスの基礎を理解し、実施できる準備を完了することを目標としている。

モデリングアセスメント

UML でモデリングを始めただけで、自分のモデリングスキルがどのくらいなのか気になったことはないでしょうか。また、リーダー、マネージャであれば、自組織のメンバーがどのくらいのモデリングスキルを持っているのか知りたいこともあると思います。しかし、そういったモデリングスキルを診断してくれるサービスは世の中にあまりないのが現状です。

そうした現状に対し、豆蔵ではモデリングスキルがどのくらい備わっているかを調べ、モデリングアセスメントを開発しました。本アセスメントでは、対象者にモデルを作成してもらい、モデルの評価を通して、その人がどのくらいのモデリングスキルを持っているのかを複数の観点から診断していきます。診断の観点は豆蔵独自の「モデリングに必要なことは何か」を分析し、考案したのになります。

このアセスメントを受けることにより、モデリングスキルとして既に身につけている部分や、モデリングスキル向上のためにどういった部分に気を付けていけば良いかが分かります。

UML モデリングのスキルアップを考えている方や、これから UML を使ったモデルベース開発を考えている方は是非、受けてみて下さい。

実 施 方 法

1. 弊社からの出題に対して制限時間(1 時間)内にモデルを作成し、提出して頂きます
2. 回答の内容を分析し、複数の観点からモデリングスキルがどの程度なのかをアセスメントします
 - アセスメントには一週間程かかります
 - アセスメントは対象者毎に行います
3. アセスメント結果と対象者の傾向を報告いたします
(※ 基本的に回答頂いたモデルは返却しませんが、ご要望があれば、添削を行った上での返却にも対応致します)

受講料 (税込)

アセスメント対象者一人あたり 66,000 円

対 象 者

- 自身の UML モデリングのスキルがどのくらいなのかを知りたい方
- 組織のメンバーの UML モデリングスキルがどのくらいなのかを知りたい方

前 提 条 件

- UML(クラス図)を使ったモデリングができること

内 容

1. はじめに

- 1) 本講座の目的
- 2) 本講座の達成目標
- 3) 分るようになって欲しいこと
- 4) 進行の目安

2. 開発プロセスの概要

- 1) 何故開発プロセスが必要なのか？
- 2) ソフトウェア開発プロセスとは？
- 3) 開発プロセスの設計
- 4) 開発プロセスの設計と計画との関係
- 5) マクロレベルのプロセス
- 6) ミクロレベルのプロセス
- 7) 代表的な OO 開発プロセス
- 8) 理解度テスト

3. RUP(Rational Unified Process)の解説

- 1) 概要
- 2) ライフサイクル
- 3) プラクティス
- 4) ロール
- 5) 作業成果物
- 6) 開発の進め方
- 7) Waterfall との違い
- 8) 理解度テスト

4. XP(eXtreme Programming)の解説

- 1) 概要
- 2) ライフサイクル
- 3) プラクティス
- 4) ロール
- 5) 作業成果物
- 6) 開発の進め方
- 7) RUP との違い
- 8) 理解度テスト

5. 開発プロセスの選定方法

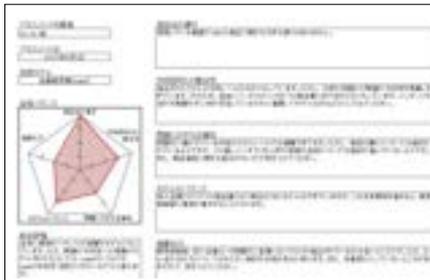
- 1) 開発プロセスのタイプ
- 2) 選定するための重要要因
- 3) 選定の進め方
- 4) サンプル

内 容

【実施方法】

1. 弊社からの出題に対して制限時間(1 時間)内にモデルを作成し、提出して頂きます
2. 回答の内容を分析し、複数の観点からモデリングスキルがどの程度なのかをアセスメントします
 - ・ アセスメントには一週間程かかります
 - ・ アセスメントは対象者毎に行います
3. アセスメント結果と対象者の傾向を報告いたします
(※基本的に回答頂いたモデルは返却しませんが、ご要望があれば、添削を行った上での返却にも対応致します)

【アセスメント例】



プロジェクト体験型演習講座

ソフトウェア開発を成功に導くには、プログラミング技術だけでなく、エンジニアリングの観点から開発工程全体を理解し、実践できることが不可欠です。

本講座「プロジェクト体験型演習講座」では、Arduino を使って 組み込みソフトウェアの基本とソフトウェアエンジニアリングの基本を学びます。要件定義から設計・実装・テストまでの工程を順に体験し、ブレッドボード・センサ・LED などの実機を用いた演習を通じて、座学で得た知識をすぐに実践に活かせる構成となっています。単なる流れの模倣ではなく、各工程の意味や本質を理解しながら進められる点が大きな特徴です。

一般的な体験型講座は「全体を通す体験」に留まりがちですが、本講座では「なぜその工程が必要か」「どう考えれば次に進めるか」を重視し、エンジニアリングの理解を深めることに重点を置いています。これにより、活動をなぞるだけでは得られない、実務直結の知識とスキルを習得できます。

さらに、本講座は チーム開発の体験 を重視しています。役割分担や情報共有を通じてコミュニケーションを取りながら開発を進めることで、実務さながらの協働を経験できます。エンジニアリングを踏まえつつ、人との協力を伴った開発を学べる講座は非常に珍しく、受講者にとって大きな価値となります。

本講座は最大で 50 名規模まで対応可能 です。実機を使った演習とチーム開発体験を通じ、参加者は開発プロセスの全体像と工程ごとの意味を理解すると同時に、実務で活かせる力を短期間で確実に身につけられます。新入社員教育や若手エンジニア育成に最適な、他に類を見ない研修です。

日 数

3 日

時 間

6 時間 / 日 (10:00~17:00)

受 講 料 (税 込)

1,650,000 円

- ※オンラインでの開催となります。
- ※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- ソフトウェア開発の流れを実践的に理解したい初心者～中級者の開発者
- システム開発の基本を体系的に学びたい新入社員・若手エンジニア

前 提 条 件

- 基本的なプログラミング知識(C言語)
- UMLの知識は不要です

別 達 目 標

- なぜソフトウェア開発においてエンジニアリングが重要なのかを理解する
- ソフトウェア開発の基本的な工程と各工程での活動を理解する
 - 要求(分析と要件決定)～設計～実装～テスト
- チーム開発を体験する
 - 他のメンバーとコミュニケーションを取りながらの開発の体験

システムズエンジニアリング入門講座

製品開発/システム開発は年々、難しいものになってきています。その背景には、システムの高度化・高性能化・多機能化、安全性・セキュリティへの配慮、会社をまたがった複数システムの連携によるサービスの実現といった様々な状況が関係しています。そうした製品やシステムの開発をスムーズに進めていくのは難しく、また、開発の途中で間違いに気がついた際の手戻りによる、時間やコストの損失も大きなものになっています。

そうした現状において、製品開発/システム開発を円滑に進めていくための技術として、システムズエンジニアリングが注目されています。また、ソフトウェア開発におけるモデルベース開発(Model Based Development)の普及と相まって、モデルを活用したシステムズエンジニアリングである、MBSE(Model Based Systems Engineering)も注目されています。

本講座では、このシステムズエンジニアリングや MBSE とはどのようなものなのか、どのような効果があるのか、そして、具体的な活動や実施する際の注意点等について取り上げていきます。また、MBSE と関連して取り上げられる SysML と MBSE の関係についても説明していきます。

本講座の講師は実際にシステムズエンジニアリングの導入を支援しているコンサルタントが行います。そのため、教科書的な内容ではなく、実際に現場でどのような課題があるのかを踏まえた講義内容になります。システムズエンジニアリングとはどのようなものかを素早く知りたい、これからシステムズエンジニアリング・MBSE の導入を始めるという方に最適な講座となっています。

また、本講座の内容を基盤としつつ、システムズエンジニアリングをより概論レベルにまとめ、MBSE の理解を補強して SysML 講座へとスムーズにつなげられるように構成した「MBSE 向け システムズエンジニアリング入門講座」もご用意しています。MBSE に強い関心をお持ちの方や、SysML 講座とあわせて受講される方に特にオススメです。

日 数

1 日

時 間

6 時間 (10:00~17:00)、MBSE 向け 6.5 時間 (10:00~17:30)

受 講 料 (税 込)

550,000 円

- ※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
- ※オンライン・オンライン (Zoom 等のオンライン会議システムを利用して実施可能)とも同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- システムズエンジニアリング・MBSE とは何かを知りたい人
- これからシステム領域の活動に携わる予定の人
- SysML を使ってみたい人/使っている人

前 提 条 件

- システム開発(製品開発)の経験者(ハードウェア開発、ソフトウェア開発は問いません)

別 達 目 標

- システムズエンジニアリングを行うことのメリットが理解できるようになります。
- システムズエンジニアリングを構成する各活動の目的/役割、主な実施事項、作業成果物が理解できるようになります。
- システムズエンジニアリングにおけるモデルの活用方法が理解できるようになります。

内 容

- はじめに
- 業務としてのソフトウェア開発
 - 2.1 顧客
 - 2.2 品質
 - 2.3 規模
 - 2.4 納期
 - 2.5 相反する制約
 - 2.6 エンジニアリングの必要性
- 組込ソフトウェア
 - 3.1 組込ソフトウェアとは
 - 3.2 組込ソフトウェアの歴史
- 組込ソフトウェアの基本
 - 4.1 基本的な考え方
 - 4.2 Arduino
 - 4.3 組込ソフトウェアプログラミングの基本
- 要求エンジニアリング
 - 5.1 要求エンジニアリングの重要性
 - 5.2 要求エンジニアリングに出でくる用語
 - 5.3 要求の特徴
 - 5.4 要求エンジニアリングの全体像
 - 5.5 要求分析
 - 5.6 ユースケースを使った要求分析
 - 5.7 ユースケースの効果
- 要求分析演習
 - 6.1 今回の題材:衝突警告装置 成果発表について
- 設計とは
 - 7.1 なぜ設計が必要なのか
 - 7.2 設計とは?
 - 7.3 ソフトウェア設計の階層参考資料:エンジニアリング活動の基本構造
- 設計演習
 - 8.1 設計演習設計演習:設計課題の抽出
設計演習:解決方法の検討
- 実装とは
 - 9.1 実装の目的
 - 9.2 主な活動
 - 9.3 設計およびテストとの関連
 - 9.4 ソフトウェア構築の基本原則
 - 9.5 プログラム品質に対する意識
 - 9.6 テハック
- 実装演習
 - 10.1 衝突警告装置の実現
 - 10.2 衝突警告装置の機能
 - 10.3 超音波センサの仕様
 - 10.4 実装のポイント実現の難度
- 10.5 考察資料

内 容

- はじめに
 - 1) 本講座の目的と到達目標
 - 2) 受講対象者と前提条件
 - 3) 進行の目安
- システムズエンジニアリングの有効性**
システムズエンジニアリングの全体像を理解します。
 - 1) 現状の課題
 - 2) システムズエンジニアリングとは
 - 3) 効果
 - 4) システムズエンジニアリングの現状
 - 5) リファレンス
- システムズエンジニアリングの活動**
システムズエンジニアリングを構成する具体的な活動について理解します。
 - 1) システムズエンジニアリングの活動の全体像
 - 2) システムズエンジニアリングの活動の要点
 - 3) システムズエンジニアリングの各活動の詳細
 - 4) テーラリング
 - 5) プロセス事例
- MBSE**
MBSE の概要と有効性について理解します。
 - 1) MBSE とは
 - 2) MBSE の有効性
 - 3) モデルとは
 - 4) モデルの種類
 - 5) モデリング言語
 - 6) SysML
 - 7) MBSE の実施にあたっての注意点

SysML 入門講座 (SysML v1)

オブジェクト指向におけるモデル表記法である UML (Unified Modeling Language) は、1997 年 11 月のバージョン 1.1 公開の後着実に普及し、オブジェクト指向でのソフトウェア・エンジニアリングの分野では今や世界規模でデファクトスタンダードの地位を確立しています。

UML の標準化を行っている OMG (Object Management Group) では、INCOSE (International Council on Systems Engineering) などの団体と協力し、ハードウェア / 電気回路 / ソフトウェアを組み合わせてシステム構築を行うシステムズ・エンジニアリング分野に向けたモデル表記法の制定 / 標準化も推進しています。そのような活動の成果としてまとめられたものが **OMG SysML (OMG Systems Modeling Language)** です。

SysML は (UML と同様) モデルを表現するために用いられる各種ダイアグラムの表記法とその意味付けを定義したもので、できるだけ UML を流用するというコンセプトで標準化されています。

業界動向としても、Lockheed Martin 社や IBM 社等が中心となって適用事例を増やしている他、SysML をベースとしている EAST-ADL2 は、車載電子システム開発向けのアーキテクチャ記述言語として注目を集めています。また、OMG と INCOSE が共同で SysML 関連の資格認定プログラム(OCSPM)を開発するなど、システムズ・エンジニアリング分野における標準的な技術要素のひとつとして普及していく可能性が高いと注目されはじめています。

反面、SysML はボリュームも大きく(たとえば、SysML で使用されるダイアグラムの種類だけで 9 種類もあります)、残念ながら独学での導入学習はかなり数層が高くなってしまっています。

本講座は、システム開発に対して SysML を適用していく際に前提となる必須の知識(SysML の概要と各ダイアグラムの主要な表記法)を 1 日の講座で効率的に習得できるように、重要なポイントに絞った講義と簡単なシステム開発を題材としたいくつかの小演習によって構成されています。また、最新版(2017 年 5 月現在)である SysML1.5 に対応しています。これによって、実際のシステム開発に SysML を導入するための足掛かりを得ることができるようになっています。

※ 本講座は UML の事前知識をお持ちでない方でも受講していただけます

日 数

1 日

時 間

7 時間(9:00~17:00)

受 講 料 (税 込)

550,000 円

※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom 等のオンライン会議システム)を利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- ハードウェア / 電気回路 / ソフトウェアの組み合わせで構成されるシステムの開発に携わる方。
製品企画担当者、システムアーキテクト、ハードウェア技術者、電気回路技術者、ソフトウェア技術者、など
- SysML の概要を知りたい方。

前 提 条 件

- ハードウェア設計、電気回路設計、ソフトウェア設計のいずれかの分野で数年以上の実務経験をお持ちの方。
※ UML に関する事前知識は特に前提としません。

SysML モデリング演習講座 (SysML v1)

UML や SysML などのモデリング言語は、表記法(読み/書きの仕方)だけでなく、その使い方を覚えて初めて効果を得ることができます。また、モデリング・スキルは多くのモデルを読み/書きし、レビューを受けなければなかなか身に付かない技術でもあります。「それはわかっているけど、そんな機会、なかなか無いんだよ」とお嘆きの方も多いでしょう。この講座はそうした方々にぴったりの講座です。

本講座は、システム工学の基本的な概念と SysML 表記法を習得した人が、「知識」として得た情報を「モデリング・スキル」として定着/使いこなせるようになるための「基礎訓練」の講座です。ある仮想的なシステムを題材にして、1 日を通して SysML の主要なダイアグラムを使ったモデリング演習とレビューを繰り返していきます。「SysML 入門講座」の方は SysML の表記法を覚えることにフォーカスしてモデルそのものの良し悪しやモデリングする際の考え方などには触れていないのに対して、本講座では「良いモデルを作っていくためにはどうすれば良いか」という方向にもフォーカスしていきます。

日 数

1 日

時 間

7 時間(9:00~17:00)

受 講 料 (税 込)

550,000 円

※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom 等のオンライン会議システム)を利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- ハードウェア / 電気回路 / ソフトウェアの組み合わせで構成されるシステムの開発に携わる方(製品企画担当者、システムアーキテクト、ハードウェア技術者、電気回路技術者、ソフトウェア技術者、など)

前 提 条 件

- ハードウェア設計、電気回路設計、ソフトウェア設計のいずれかの分野で数年以上の実務経験をお持ちの方
- システム工学の基本的な概念を習得している方
- SysML の基本的な表記法を習得している方

訓 練 目 標

- モデル要素を正確に適用した SysML のダイアグラムが描けるようになる。
- 複数のダイアグラムを組み合わせても、矛盾が生じないモデルを作れるようになる。
- ハードウェア/電気回路/ソフトウェアの組み合わせによって構成されるシステム・レベルのモデリングで気をつけなければならないことを理解し、人に説明できるようになる。

訓 練 目 標

- SysML の背景、概要、採用のメリットを人に説明できるようになります。
- SysML の学習した範囲内で記述されたモデルであれば(ie 大方のモデルが)理解できるようになります。
- 学習した SysML のダイアグラムのいくつかを現場で使い始められるようになります。

内 容

形式:講義+演習 (個人/グループ)

1. はじめに
2. SysML 概要
 - 1) SysML とは
 - 2) SysML 登場の背景
 - 3) UML でできないこと
 - 4) SysML の歴史
 - 5) UML との関係
 - 6) SysML の効果
 - 7) SysML を理解するためのポイント
 - 8) SysML でできること
 - 9) 事例
 - 10) SysML の今後
3. 各図で共通の要素
4. パッケージ図
5. 要求図
6. ユースケース図
7. ブロック定義図
8. 内部ブロック図
9. パラメトリック図
10. アクティビティ図
11. シーケンス図
12. ステートマシン図
13. おわりに

内 容

形式:講義+演習 (個人/グループ)

1. はじめに
2. モデリングとは?
3. システムズ・エンジニアリングとモデリング
4. システム要求のモデリング
5. 静的構造のモデリング
6. 動的構造のモデリング
7. 機能的構造のモデリング
8. モデリングの共通留意事項
9. おわりに

SysML v2 入門講座

近年、システム開発の大規模化・複雑化や SoS(System of Systems)・IoT への拡張に伴い、システムズ・エンジニアリングを支えるモデリング技術の重要性が一層高まっています。その中心にあるのが、OMG により刷新された SysML v2 です。SysML v2 は、システムの図の中にメカ・エレキ・ソフトすべての要素を統合的に表現でき、システム全体の総合的な検討を可能にします。

本講座のテキストは、UML や SysML v1 未経験の方でも理解できるように、それらの知識を前提とせずに構成しています。これからシステムのモデリングを始める方を意識して、SysML v2 の基本概念、要求・構造・振る舞いといった主要な表記法を体系的に学びます。また、講義と演習を組み合わせ、座学での理解不足に気づきながら学習を深められるのも特長です。

一日の講座を通じて、SysML v2 の表記法を理解し、読み書きの基礎を学びます。

- **システムエンジニア**: システムのモデルを描き始められるようになり、分析・設計力を強化できます。また、他のシステムエンジニアとの意見交換する力を高めます。
- **上流設計者(メカ・エレキ・ソフト)**: システムエンジニアの SysML v2 モデルを正しく解釈し、自分の考えを明確に伝える力を獲得し、設計者との橋渡し役を果たせます。
- **製品企画担当者**: エンジニアの SysML v2 モデルを正しく解釈し、エンジニアとともに、要求や構造を深く検討できるようになります。

日 数

1 日

時 間

7 時間(9:00~17:00)

受 講 料 (税 込)

550,000 円

- ※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
- ※オンライン(Zoom 等のオンライン会議システムを利用して実施可能)とも同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- ハードウェア/電気回路/ソフトウェアの組み合わせで構成されるシステムの開発に携わる方。
製品企画担当者、システムアーキテクト、ハードウェア技術者、電気回路技術者、ソフトウェア技術者、など
- SysML の概要を知りたい方。

前 提 条 件

- ハードウェア設計、電気回路設計、ソフトウェア設計のいずれかの分野で数年以上の実務経験をお持ちの方。

副 達 目 標

- SysML の背景、概要、採用のメリットを人に説明できるようになります。
- SysML の学習した範囲内で記述されたモデルであれば(ie 大方のモデルが)理解できるようになります。
- 学習した SysML のダイアグラムのいくつかを現場で使い始められるようになります。

SysML v2 モデリング演習講座

SysML v2 モデリング演習講座

UML や SysML などのモデリング言語は、表記法(読み/書きの仕方)だけでなく、その使い方を覚えて初めて効果を得ることができます。また、モデリング・スキルは多くのモデルを読み/書きし、レビューを受けなければなかなか身に付かない技術でもあります。「それはわかっているけど、そんな機会、なかなか無いんだよ」とお嘆きの方も多いでしょう。この講座はそうした方々にぴったりの講座です。

本講座は、システム工学の基本的な概念と SysML 表記法を習得した人が、「知識」として得た情報を「モデリング・スキル」として定着/使いこなせるようになるための「基礎訓練」の講座です。ある仮想的なシステムを題材にして、1 日を通して SysML の主要なビューを使ったモデリング演習とレビューを繰り返していきます。「SysML v2 入門講座」の方は SysML の表記法を覚えることにフォーカスしてモデルそのものの良い悪いやモデリングする際の考え方などには触れていないのに対して、本講座では「良いモデルを作っていくためにはどうすれば良いか」という方向にもフォーカスしていきます。

日 数

1 日

時 間

7 時間(9:00~17:00)

受 講 料 (税 込)

550,000 円

- ※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
- ※オンライン(Zoom 等のオンライン会議システムを利用して実施可能)とも同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- ハードウェア/電気回路/ソフトウェアの組み合わせで構成されるシステムの開発に携わる方(製品企画担当者、システムアーキテクト、ハードウェア技術者、電気回路技術者、ソフトウェア技術者、など)

前 提 条 件

- ハードウェア設計、電気回路設計、ソフトウェア設計のいずれかの分野で数年以上の実務経験をお持ちの方
- システム工学の基本的な概念を習得している方
- SysML v2 の基本的な表記法を習得している方

副 達 目 標

- モデル要素を正確に適用した SysML v2 のビューが描けるようになる。
- 複数のビューを組み合わせても、矛盾が生じないモデルを作れるようになる。
- ハードウェア/電気回路/ソフトウェアの組み合わせによって構成されるシステム・レベルのモデリングで気をつけなければならないことを理解し、人に説明できるようになる。

内 容

形式: 講義+演習 (個人/グループ)

1. はじめに
2. SysML 概要
3. SysML v2 の基本表現
4. 要求の表現
5. 構造の表現
6. アクションの表現
7. 状態の表現
8. シーケンスの表現
9. 制約の表現
10. パリエーションの表現
11. パッケージの表現
12. 割り当ての表現
13. おわりに
14. 付録

内 容

形式: 講義+演習 (個人/グループ)

1. はじめに
2. モデリングとは?
3. システム工学とモデリング
4. システム要求のモデリング
5. 静的構造のモデリング
6. 動的構造のモデリング
7. 機能的構造のモデリング
8. モデリングの共通留意事項
9. おわりに

MBSE 基礎講座(システム設計編)

近年、製品/システムの高機能化、多機能化が顕著になってきています。それに伴い、開発の大規模化、システムの複雑化が進み、そうしたシステムをどのように効率的に開発していくのかが大きな課題になっています。この問題を解決するための技術のひとつとして、モデルを活用したシステムズエンジニアリング、MBSE(Model Based Systems Engineering)が注目されています。

本講座では MBSE を実践できるようにするために、製品開発/システム開発における活動を MBSE ではどのように進めていくのかについて、要求エンジニアリングとシステム設計の観点から説明していきます。

要求エンジニアリングではシステムに対する要求獲得、要求分析、要件作成を行っていく中で、SysML をどのように活用していくか、システム設計では設計課題の抽出、システム構造の記述、システムの振舞いの記述、トレードオフ分析といった活動の中で SysML をどのように活用するのかを、具体的にモデルを作成しながら学習していきます。

豆蔵は SysML がバージョン 1.0 になる前から、いち早く MBSE に着目し、実践してきました。本講座にはその過程で得られたノウハウが詰まっており、教科書的ではなく、実践的な内容になっています。これから MBSE を始めようとしている方、MBSE に関する具体的な情報が困っている方にピッタリの講座となっていますので、是非、受講してみてください。

日 数

1 日

時 間

7 時間(9:00~17:00)

受 講 料 (税 込)

550,000 円

※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- これから MBSE やシステムズエンジニアリングを始めようとしている方
- システムズエンジニアリングに関する具体的な情報を収集している方

前 提 条 件

- SysML に関する基本的な知識
- 製品開発の経験

例 達 目 標

- SysML を利用した MBSE の進め方を理解し、他者に説明できる
 - MBSE における要求エンジニアリングの活動の理解
 - MBSE におけるシステム設計の理解
- MBSE における SysML のモデルを作成できる

システムアシュアランス入門講座

システムが社会基盤に深く関与するようになって、システムの安全性やセキュリティがより問われるようになってきています。こうした状況の変化にもなって、システムを開発する際、利害関係者に「どうしてこのシステムは安全であるといえるのか」や「開発の進め方に問題がないこと」を説明することも求められるようになってきました。また、安全規格やセキュリティ規格には、そうした説明を必須としているものもあります。

こうした、安全であることやセキュアであることを他者が理解できるように説明し、その内容について議論するのは難しい問題でした。この問題に対し、近年、システムアシュアランスの技術が注目されています。システムアシュアランスは上記のような安全性やセキュリティに関する説明(それだけではありません)を網羅的に、また、議論しやすい形で表現する技術です。

本講座ではこのシステムアシュアランスの技術を取り上げ、その考え方や使い方についてレクチャーしていきます。安全規格対応のみならず、自身の考え方を他者にきちんと説明したいという方に最適な講座となっています。

日 数

1 日

時 間

6 時間(10:00~17:00)

受 講 料 (税 込)

550,000 円

※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- システムの安全性やセキュリティについて他者に説明する必要がある方
- 業務でセーフティケースやアシュアランスケースを書く必要のある方

前 提 条 件

- ソフトウェア開発、若しくは、製品開発の経験があるのが望ましい

例 達 目 標

- システムアシュアランスの必要性、意義を他者に説明できるようになる
- GSN を用いて、アシュアランスケースを作成できるようになる

内 容

- はじめに
 - 1.1 本講座の目的と到達目標
 - 1.2 受講対象者と前提条件
 - 1.3 進行の目安
- システムズエンジニアリングにおけるシステム設計
 - 2.1 ISO15288 におけるシステム設計
 - 2.2 OOSEM におけるシステム設計
 - 2.3 システム設計とは
- システム設計の活動
 - 3.1 活動の全体像
 - 3.2 システム要件の決定
 - 3.3 設計の基本手順
 - 3.3.1 評価基準の設定
 - 3.3.2 解決策の検討
 - 3.3.3 解決策の評価・決定
 - 3.3.4 アーキテクチャ構造の検討
 - 3.3.5 トレース
 - 3.4 論理アーキテクチャと物理アーキテクチャの違い
- 活動の詳細
 - 4.1 設計課題の抽出
 - 4.2 設計課題の優先順位付け
 - 4.3 論理アーキテクチャ設計
 - 4.3.1 論理アーキテクチャで解決する設計課題
 - 4.3.2 システム分割(静的構造の検討)
 - 4.3.3 振舞いの検討(動的・機能的構造の検討)
 - 4.3.4 システム構成要素への機能の割当て
 - 4.3.5 アーキテクチャ構造の決定
 - 4.3.6 設計課題解決の確認
 - 4.4 物理アーキテクチャ設計
 - 4.4.1 物理アーキテクチャで解決する設計課題
 - 4.4.2 物理アーキテクチャの構造の検討
 - 4.4.2.1 協調設計
 - 4.4.3 設計課題解決の確認

内 容

- はじめに
- システムアシュアランス概要
 - 2.1 現状の課題
 - 2.2 システムアシュアランスとは
 - 2.3 アシュアランスケースの効果
 - 2.4 アシュアランスケースの記述方法
- アシュアランスケースの表記
 - 3.1 議論モデル
 - 3.2 ゴール(Goal)
 - 3.3 ストラテジ(Strategy)
 - 3.4 エビデンス(Evidence)
 - 3.5 コンテキスト(Context)
 - 3.6 アンチベロウズ(Undeveloped)
 - 3.7 リンク(Link)
 - 3.8 その他の要素
 - 3.9 D-Case での追加要素
- アシュアランスケースの作成
 - 4.1 アシュアランスケースの作成手順
 - 4.2 アシュアランスケースの妥当性評価
- アシュアランスケースを使った合意形成
- 関連技術
 - 6.1 議論パターン
 - 6.2 DAF
- モデルベース開発との連携

アジャイル入門講座

ソフトウェアの歴史は 40 年とも 50 年とも言われます。その間にソフトウェアはビジネスに不可欠な要素となり、ビジネスの変化の影響を強く受けるようになりました。かつては、開発当初に一度決めた仕様は開発中にほとんど変更を受けることはありませんでしたが、ビジネス状況がめまぐるしく変化する現代においては、ソフトウェアはその本来の柔軟（=ソフト）さを最大限に発揮する必要性に迫られています。

アジャイル開発は、こういった世の中のニーズに応えるため、できるだけ小さな「動作するソフトウェア」を短期間で作成し、これを繰り返す（反復）ことで最終的な完成品を得るという考え方で、反復毎に新たな要件や要件の変更を受け容れていけるのが大きな特徴です。

ですが、アジャイル開発はその源流となる考え方が従来の計画重視型のプロセスとは大きく異なるため誤解も多く、従来の手法からの移行が非常に困難で、普及の妨げとなったり、導入しても効果が得られないなどのような状態となっているのも現実です。

本講座では、アジャイル開発とは何か、どういう意義があるのか、どのような場合にアジャイル開発を採用する必要があるのかを理解することを目的としています。表面的なプロセスの説明では分かりづらい「なぜそうするのか」という説明に注力し、グループディスカッションと演習で理解を深めます。

日 数

1 日

時 間

7時間(9:00~17:00)

受講料 (税込)

550,000 円

※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。

※オンサイト・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

アジャイル開発についてその意義を理解したいエンジニア・管理者

アジャイル開発を始める際にメンバー全員の意識統一をしたい方

前 提 条 件

- 一般的なソフトウェア開発プロセス(ウォーターフォール開発)の理解
- ソフトウェア開発の経験

到達 目 標

- アジャイル開発の意義を理解できる。
- 基本的なアジャイル開発プロセスを理解できる。
- 一般的なアジャイル開発の用語を理解できる。

スクラム基礎と実践講座

ここ 30~40 年の間にビジネスと IT は密接に絡み合いながら急速な発展を遂げました。その過程で、ウォーターフォール型の従来の開発手法とは別に、開発に携わる人にフォーカスした軽量な開発手法が生み出されました。それがアジャイル開発手法です。欧米では、すでにウォーターフォールは過去のものであり、アジャイル開発が主流となっています。アジャイルは反復(イテレーション)と呼ばれる短い開発期間単位で開発を進め、早期に要求や技術的なリスクを最小化する手法です。その評価を受けて改良することは、プログラムの価値を高める一番の近道であり、結果的に生産性や品質の面でも有効であると考えられているからです。

本講座は、アジャイル開発手法の中でも主流のスクラムをとりあげ、アジャイルの生まれた背景や基本知識、開発の進め方をわかりやすく解説し、演習を通じて理解を深めることによってアジャイル開発の全体像を理解して頂きます。

日 数

1 日

時 間

7時間(9:00~17:00)

受講料 (税込)

550,000 円

※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。

※オンサイト・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- アジャイル開発についてその意義を理解したいエンジニア・管理者
- アジャイル開発を始める際にメンバー全員の意識統一をしたい方

前 提 条 件

- ウォーターフォールなど従来のシステム開発経験者の方
 - なぜアジャイル開発が必要かというアジャイルの意義を理解している
- ※弊社トレーニング「アジャイル入門」修了程度の知識

到達 目 標

- アジャイル開発における基礎知識が理解できる。
- スクラムを利用したシステム開発の進め方が理解できる。

内 容

- 1 アジャイルとは
 - 1.1 アジャイルソフトウェア開発プロセスとは
 - 1.2 アジャイルソフトウェア開発宣言
 - 1.3 アジャイル開発12の原則
 - 1.4 さまざまなアジャイル開発手法
 - 1.5 よく使われるアジャイル手法
 - 1.6 スクラム
 - 1.7 エクストリーム・プログラミング
- 2 アジャイルの特徴
 - 2.1 適応型
 - 2.1.1 ウォーターフォールの前提
 - 2.1.2 アジャイルの前提
 - 2.2 人が中心
 - 2.3 反復開発
 - 2.3.1 計画とコントロール
 - 2.4 スコープ操作
 - 2.4.1 使われないシステム
 - 2.5 コミュニケーションの変化
 - 2.6 進化的設計
- 3 基本的なアジャイルプロセス
 - 3.1 反復のサイクル
 - 3.2 ロール
 - 3.3 会議体
 - 3.4 ストーリーとタスク
 - 3.5 ストーリーの見積もり方法
 - 3.6 用語の対応
- 演習 ディスカッション 1/2
4. アジャイルプラクティス概要
 - 4.1 プラクティスとは
 - 4.2 タスクボード
 - 4.3 バーンダウンチャート
 - 4.4 構成管理
 - 4.5 継続的インテグレーション
 - 4.6 テスト駆動開発
 - 4.7 リファクタリング
 - 4.8 ペアプログラミング
5. 演習:アジャイル開発体験

内 容

1. アジャイル開発の背景
 - 1.1 アジャイルソフトウェア開発とは
 - 1.2 ソフトウェア開発手法の変遷
 - 1.3 ウォーミングアップ(ウォーターフォールの振り返り)
 - 1.4 使われないシステム
 - 1.5 プロセスの比較
 - 1.6 ウォーターフォールの前提
 - 1.7 アジャイルの前提
 - 1.8 見積もり観点の変化
 - 1.9 実施方針の変化
 - 1.10 コミュニケーションの変化
 - 1.11 リスクコントロールの変化
 - 1.12 アジャイルマニフェスト
 - 1.13 アジャイル開発 12 の原則
 - 1.14 プロセスから人へ
 - 1.15 あらためて「アジャイル」とは
 - 1.16 まとめ
2. スクラムの基本
 - 2.1 アジャイル開発の全体像
 - 2.2 アジャイル開発手法の関係
 - 2.3 スクラムとは
 - 2.4 スクラムのプロセスフレームワーク
 - 2.5 スクラムの構造
 - 2.6 スクラムの登場人物
 - 2.7 スクラムを支えるツール
 - 2.8 必要最小限のイベント
3. スクラムの実践
 - 3.1 スクラム実践のポイント
 - 3.2 プロダクトオーナーとチーム
 - 3.3 プロダクトバックログ作成
 - 3.4 スプリントプランニング
 - 3.5 日々の作業、成果のレビュー、振り返り
 - 3.6 プロダクトバックログ調整・見直し
 - 3.7 正しいスクラム
 - 3.8 正しくないスクラム
- 4 最後に
 - 4.1 最適化と適応
 - 4.2 ウォーターフォールとアジャイルの適性
- 4.3 Scrum のプロジェクト運営

アジャイル開発のためのプロジェクト管理講座 基礎編

— 要件管理・計画・進捗管理 —

近年アジャイル開発は急速に普及していますが、その一方で「スクラムの枠に当てはめて、定期的なミーティングやバックログ管理だけで進めればよい」と誤解され、混乱を招くプロジェクトも少なくありません。実際には、アジャイル開発も従来のプロジェクト管理と同様に、要件管理・計画・進捗管理といったエンジニアリングの基本を踏まえて運営することが不可欠です。

本講座では、CMMIやPMBOKで示される従来の知識体系に基づくプロジェクト管理上の要点を、スクラムのイベントやプロダクトバックログなどアジャイル特有のプラクティスに照らし合わせて解説します。その上で、アジャイルチームメンバーで演習チームを組み、現場の状況と講義内容を照らし合わせながら議論し、講師との質疑応答を交えて理解を深めます。

アジャイル開発の実践で、これまでの開発と変わらず進捗のコントロールに悩んでいる方、要件変化に振り回されてチームが混乱している方、あるいはアジャイルを導入したものの期待した成果につながらない方にご受講をお勧めします。

本講座を通じて、アジャイル開発を単なるイベント運営にとどめるのではなく、エンジニアリングの原則に根差した「計画性と柔軟性の両立」を実現する具体的なアプローチを習得できます。これにより、チームや組織全体の開発プロセスを安定させ、スピードと品質の双方を高める実践力を獲得できます。

日 数

1日

時 間

7.5時間(9:00～17:30)

受 講 料 (税 込)

550,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※オンライン・オフライン(ZOOM等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- システム開発経験があり、手戻りの課題を感じたことがある方
- ハードウェア/ソフトウェアいずれの設計者でも参加可能
- アジャイル開発の実践者、またはこれから実践予定の方(プロダクトオーナーを含む)

前 提 条 件

- スクラムの基本的な知識があること(プロダクトバックログ・スプリント・デイリースクラム・スプリントレビュー・スプリントバックログなど)。アジャイル開発の基礎を事前に学習していると、より効果的に理解を深められます

仕様記述実践講座

— 実践的な日本語の使い方から図による記述方法まで —

ソフトウェア開発において、仕様書は非常に大きなウェイトを占める成果物です。しかし、大事なものは分かっても実際の現場で、きちんと書かれていることは非常に稀です。内容に矛盾があったり、抜けがあったりする仕様書に悩まされている開発者も多いことと思います。実際、仕様書の不完全さによる設計、実装の問題とそれによる手戻りは、ソフトウェア開発における大きな問題のひとつです。

我々のコンサルティングの現場でも、良い仕様書を作成することは普遍的な改善ポイントのひとつとして、常に挙げられています。また、オフショア開発等の外注利用においても、その成否を分けるポイントとして仕様書は重要な文書になります。

本講座ではそうした「仕様」をいかに記述するかを中心に上げ、網羅的に、正確に仕様を記述するための手順や文書のフォーマット、表記方法、考え方のポイントといった項目を説明していきます。また、本講座の大きな特徴として、そうした記述を行うためにUML(ステートマシン図)を利用していることが挙げられます。仕様記述という、自然言語や形式的なものが主流ですが、UMLも仕様記述に使えることが、この講座を通してご理解頂けると思います。

日本語での仕様記述方法を上達させたい方、より正確な仕様の記述方法を習得したい方、ユースケース記述と仕様書の違いが良く分からない方等々、仕様記述に関係する多くの方に本講座は有効です。また、演習を通して学んでいきますので、受講後、現場ですぐに試すことができるのも大きな特徴です。

仕様に関まされることが多い方、是非、受講してみてください。

日 数

1日

時 間

6時間(10:00～17:00)

受 講 料 (税 込)

550,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※オンライン・オフライン(ZOOM等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

ソフトウェア開発者、要求管理担当者、製品企画担当者、システムを発注する側の要求担当者

前 提 条 件

特になし

例 達 目 標

- 要件や仕様の記述における日本語の使用に関し、より正確な記述を心がけるようになる or ができるようになる
- 仕様記述の基本構成や要点を理解し、現行の仕様書の改善点を上げることができる
- UMLのステートマシン図が読み書きできるようになる
- 講座で習った仕様の書き方を現場で実践できるようになる

例 達 目 標

- 要件管理・計画・進捗管理の各要点を理解し、自身のアジャイルチームに取り入れると良いところが明らかになる。
- 自身のアジャイルチームで、改善可能な点と手始めの改善案を検討できている

内 容

1. はじめに

2. 要件管理

- 2.1 要求エンジニアリングの基本知識
- 2.2 要件管理の目的
- 2.3 要件管理の流れ
- 2.4 管理すべき要求/要件
- 2.5 要件管理の基本(事前レビュー、変更管理、トレーサビリティ、スコープ制御)
- 2.6 演習

3. 計画

- 3.1 計画の意義
- 3.2 計画で取り上げられるべき項目
- 3.3 作成する計画の種類
- 3.4 計画作成の基本フロー
- 3.5 計画に関する全般的な留意点
- 3.6 演習

4. 進捗管理

- 4.1 進捗管理の要点
- 4.2 進捗管理を構成する活動(進捗会議、ステアリング会議、是正と再計画、公式レビュー、ポストモーターなど)
- 4.3 演習

内 容

1. はじめに

- 1)本講座の目的
- 2)到達目標
- 3)本講座の構成
- 4)講義進行の目安

2. 要件記述

- 1)要件記述のポイント
- 2)要件記述例
- 3)要件記述例解説
- 4)要件や仕様における日本語の使用

3. UMLのステートマシン図

- 1)基本状態遷移
- 2)遷移
- 3)アクティビティ
- 4)自己遷移と内部遷移
- 5)コンボジット状態
- 6)直交状態(並行状態)
- 7)履歴状態

4. 仕様記述

- 1)仕様とは
- 2)仕様記述の基本の“基”
- 3)機能の仕様記述の基本
- 4)仕様の基本骨格
- 5)仕様記述のステップ

アーキテクチャ設計講座

本講座はオブジェクト指向を使った分析～設計に関する実践講座です。
これまでもオブジェクト指向による分析・設計の教育は、弊社の教育コースも含め、数多くありました。
しかしそれらは初級クラスを対象としたものであり、オブジェクト指向それ自身の習得に焦点が当たっていました。例えば設計で焦点が当てられているのは、いかに設計モデルを作るか・いかに良い設計モデルを作るかであって、いかに適切に設計を行うかではありませんでした。また題材もやさしいものが多く、現場で遭遇する難しい問題は取り上げられてきませんでした。

これらを受け開発したのが本講座です。
特徴は、
①オブジェクト指向を使ってどう適正・健全なエンジニアリングを実現するのかに重点をおいていること
②中上級者を対象に現場で遭遇する難しい課題を題材としてとりあげステップアップを図れるメニューを用意していること、にあります。
③既存コースでとりあげられていた内容もこれまでの教育・コンサルティング経験に基づき大幅見直し・強化を図っています。

本講座はアーキテクチャ設計を隅にとりあげているのが大きな特徴になっています。

日 数

2 日

時 間

6 時間/日 (10:00～17:00)

受 講 料 (税 込)

1,100,000 円
※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
※オンサイト・オンライン (Zoom 等のオンライン会議システムを利用して実施可能) とともに同料金です。
※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- アーキテクト
- 設計者
- プロセスエンジニア (ie. SEPG)
- プロジェクト管理者

前 提 条 件

- オブジェクト指向の基本概念 (クラス、インスタンス、継承) を理解していること。
- UML の基本要素 (クラス図、シーケンス図、ステートマシン図、パッケージ図、コンポーネント図、パッケージ図、配置図) が使えること。
- オブジェクト指向設計経験があること。(必須ではない)

状態遷移設計基礎講座

さまざまな設計技法がある中、この講座では状態遷移図 (ステートマシン図) を活用する振る舞いの分析・設計にフォーカスし、多くの演習題材を交えながらソフトウェア・システムの動的側面のモデル化や網羅的なテスト設計などに関する実践的な知見を得ていきます。直接目で見るできないソフトウェアの振る舞いを状態と遷移という観点で捉えてモデル化 (見える化する) することで、より理解しやすく拡張・変更もしやすいシジの良い構造を検討できるようになる事を狙います。

この講座では、状態遷移設計に関してさまざまなトピック (状態遷移の捉え方、モデルとしての表現方法、実装とのマッピングやリバース・エンジニアリング、複数のステートマシンの連携、派生開発時の留意点、網羅的なテストケースの作成など) に触れ、小さな演習を交えながら手を動かすことで理解を深めていきます。状態遷移設計に関してここまで体系的/多角的にまとめられている資料や書籍はほとんど見当たらないので、非常に貴重・稀有な講座になっていると言えます。特に組み込みソフトウェア・システム開発において状態遷移設計は非常に重要かつ有効な設計手法なので、状態遷移設計に関して 2 日間で体系的/実践的に学べる本講座は、より良い振る舞い設計を目指したいソフトウェア設計者に強くお勧めできるものになっています。

日 数

2 日

時 間

7 時間/日 (9:00～17:00)

受 講 料 (税 込)

1,100,000 円
※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
※オンサイト・オンライン (Zoom 等のオンライン会議システムを利用して実施可能) とともに同料金です。
※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

ソフトウェア設計者

前 提 条 件

- ソフトウェア開発の経験があること
- C 言語を使ったプログラミングに関する経験、若しくは、C 言語に関する基本的な知識があること
- UML/ SysML のステートマシン図の表記法に関する知識
※ 正確なものを書けなくても良い
- 状態遷移表に関する知識
※ 正確なものを書けなくても良い

内 容

- 状態遷移設計の必要性を説明できる
- 自然言語で記述された仕様書から状態を読み取り、正しく状態遷移図 (ステートマシン図) を作成できる
「正しい」とは、表記として正しく、状態遷移図を作る上での検討に漏れがないこと
その上で、「良い」状態遷移図を作成できるようになるのが望ましい
- 状態遷移図とプログラムの対応関係を説明できる
- ステートマシンが複数存在するときとその整合性や連携を意識した設計ができる
- 状態遷移図を入力として、状態遷移を網羅するテストケースを作成できる

内 容

- オブジェクト指向開発におけるアーキテクチャ設計の重要性を説明できるようになる。
- オブジェクト指向開発におけるアーキテクチャ設計 (ie. 構造化) の考え方を説明できるようになる。
- 習った手順に準じ、オブジェクト指向開発のアーキテクチャ設計を実施できるようになる。

内 容

1. はじめに
2. システム設計とは (復習)
 - 1) システム設計プロセスの定義
 - 2) 妥当性評価
 - 3) システム設計書
3. アーキテクチャ設計の基本的な考え方
 - 1) 何故アーキテクチャが重要なのか
 - 2) オブジェクト指向とアーキテクチャ
 - 3) アーキテクトの役割
4. アーキテクチャ設計の進め方
 - 1) システム要件の決定
 - 2) システム要件書の確認
 - 3) アーキテクチャ課題の抽出
 - 4) アーキテクチャ候補の検討
 - 5) パッケージの特定と定義
 - 6) コンポーネントの特定と定義
 - 7) コンポーネントインタフェース設計
 - 8) メカニズム設計
 - 9) アーキテクチャ評価
 - 10) システム設計書の作成

内 容

形式: 講義 + 演習

- 【第 1 日目】
1. はじめに
 2. 状態遷移設計の必要性
 3. 状態遷移図を構成する要素
 4. 状態遷移設計をプログラムから理解する
 5. 状態遷移設計の進め方

- 【第 2 日目】
5. 状態遷移設計の進め方 (続き)
 6. 状態遷移図の妥当性確認
 7. ステートマシンの連携
 8. 派生開発時の検討方法
 9. ステートマシンを使ったテストケース設計

要求開発実践講座

近年、企業のプロダクト開発において、経営層の描く事業戦略と現場の開発活動との連携の無さが重要な課題となっています。現場の判断や想いだけで進めた結果、投資回収ができない製品や、予算停止に至るプロジェクトが少なくありません。

さらに、開発を終えて納品した後になっても、「その製品ではユーザーが期待した成果が得られない」といった事態がしばしば発生します。

従来のソフトウェアエンジニアリングでは、利害関係者から要求を獲得することに主眼が置かれ、ビジネスとしての価値検討はエンジニアリングの枠外とされてきました。

しかし、こうした問題を踏まえ、2015年に改訂された ISO/IEC/IEEE 15288:2015 (JIS X 0160:2018) では、ビジネス要求の分析から利害関係者のニーズ・システム要件定義、さらには運用に至るまでを含むシステムライフサイクルプロセスが規定され、その重要性が明確に示されています。

振り返れば、本来プロダクトやサービスは事業戦略の実現手段であり、経営層の意図を汲み取り、ユーザーの課題とその解決された姿を描き、それを具体的なプロダクト要件へと落とし込むことが不可欠です。この一連の構築しを担い、ISO規格でも強調されているのが「要求開発」です。

本講座では、経営層の事業戦略と利用者のニーズ・要求を明確にし、「どのような製品・サービスを提供すべきか」を整理した上で、そこから導かれる機能・非機能要求を定義する実践的手法を学びます。3日間にわたる演習では、参加者が役割の枠を超えて協働し、ビジネス課題と製品要求を結びつけるプロセスを体験します。

対象はシステム開発者に限らず、経営層、事業企画担当者、営業・サービス担当者など幅広い方々です。多様なメンバーが共に参加することで、より実践的かつ効果的に「新しい価値を生み出す要求開発」を体得できます。

日 数

3日

時 間

7時間/日(9:00~17:00)

受講料 (税込)

1,650,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)とに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- 製品企画をおこなう方
- 製品開発の上流工程(要求獲得~要求分析~要件定義)を担当される方

前 提 条 件

- 企業での実務経験(目安として5年程度)を有し、社内の他部署との連携や協働の経験がある方
- 製品開発の実務経験があれば、なお望ましい

ソフトウェアテスト実践講座

-『ソフトウェアテスト基礎講座』のテスト設計・実装に関連する実践講座 -

本講座はソフトウェアテストのテスト設計・実装(テストケース設計)に関する実践講座です。近年、テスト設計の教育は、弊社の教育コースも含め数多く開発されてきています。しかしそれらは初級クラスを対象としたものであり、テストケースの設計技法それ自身の習得に焦点が当たっていました。また題材もやさしいものが多く、現実的なテスト対象のごく一部しか扱っていませんでした。そのため、「テストケースの設計技法を適用するためにテスト分析・設計をどう行うか」といったような、現場で遭遇する難しい問題は取り上げられてきませんでした。例えば、オブジェクト指向・UMLベースで作った成果物(仕様、設計)からどうやってテストケースを導き出すかは、既存の教育講座ではあまり取り上げられていません。

これらを受け開発したのが本講座です。特徴は、①中上級者を対象に、具体的な仕様書、設計書から、どうテスト設計を進めていき、テストケース設計技法の適用までつなげるのNに重点を置いていること。②具体的な演習題材はオブジェクト指向・UMLベースの仕様・設計を使っているため、UMLベースの仕様書を基にしたテスト設計の方法を習得できることにあります。

ソフトウェアテスト基礎講座の第1部・第2部で学んだテスト設計・テスト実装の知識の実践・活用につなげたい方のみならず、オブジェクト指向・UMLベース開発におけるテストは具体的にどうするか模索している方にも最適なコースとなっております。テストでお悩みの皆さん、是非本講座をご活用ください。

日 数

2日

時 間

6時間/日(10:00~17:00)

受講料 (税込)

1,100,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)とに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

アーキテクト、開発者、テスト設計担当者

前 提 条 件

- 基礎ソフトウェア開発基礎講座Ⅰ、Ⅱの知識
- UMLで書かれたモデル(シーケンス、ステートマシン、クラス図)の知識(読むことができる)
- オブジェクト指向の基礎知識

例 達 目 標

- UML・オブジェクト指向の各テストベースに対し適切にテストを設計できる
- UML・オブジェクト指向のモデルベースのテストベースから、テストケース設計方針に基づき、各種技法を適切に使い、テストケースを構築することができる

例 達 目 標

- 要求開発の基本的な考え方、プロセス、成果物について理解できる。
- 要求の可視化方法について理解できる。
- ステークホルダー分析、課題・要求の整理、プロジェクトゴール設定の方法について理解できる。
- 業務の可視化方法について理解できる。
- ビジネスモデリングの3つのモデル(ビジネスユースケース、業務フロー、概念モデル)の「考え方」「進め方」「モデル間の関係」が理解できる。
- AsIsモデルをベースに業務改革後のToBeモデルを検討し、ゴールから評価する方法について理解できる。
- システム化範囲の導出方法と、システム化に必要な情報について理解できる。

内 容

形式: 講義+演習

【第1日目】

I 要求開発実践講座の概要

1. 要件定義の基礎知識
2. 要求開発の要旨
3. システム開発を前提にした要求開発

II ゴールモデリングの基礎講座

1. モデリングの基礎知識
2. ゴールモデリングの基礎知識
 - 2.1 ゴールモデリングの位置付け
 - 2.2 ゴールモデリングワークショップ

【第2日目】

II ゴールモデリングの基礎講座(続き)

- 2.2 ゴールモデリングワークショップ(続き)

III ビジネスモデリングの基礎講座

1. ビジネスモデリングの基礎知識
 - 1.1 ビジネスモデリング
 - 1.2 サービスモデリング
 - 1.3 プロセスモデリング

【第3日目】

III ビジネスモデリングの基礎講座(続き)

- 1.3 プロセスモデリング(続き)
- 1.4 情報モデリング
- 1.5 あるべき業務モデルの設計

IV システム開発へのつながり

1. システム開発へのつながり
2. まとめ(振り返り)

内 容

1. はじめに
2. 対象とする開発の概要
 - 1) 基本的な開発プロセスの骨格
3. 典型的なテストレベルとテストベースの構成
 - 1) 典型的なテストレベル
 - 2) 典型的なテストベースの構成
4. 単体テスト
 - 1) 単体テストの概要
 - 2) 単体テストのテストケース設計例
 - 3) 演習問題
5. 統合テスト I
 - 1) 統合テスト I の概要
 - 2) 統合テスト I のテストケース設計実践例
 - 3) 演習問題
6. 統合テスト II
 - 1) 統合テスト II の概要
 - 2) 統合テスト II のテストケース設計例
 - 3) 演習問題
7. システムテスト
 - 1) システムテストの概要
 - 2) テストケース設計例
 - 3) 演習問題

【ソフトウェア品質講座】レビュー実践講座

従来、レビューに関する講座というと、レビューの進め方・技法(ex. インスペクションとは)・心得を教えるものというのが相場でした。確かにそれらはレビューの基本なのですが、残念ながらそれらを理解/習得したからといって各人のレビュー力が向上するわけではありません。

過去、レビュー力は日々の業務の中のレビューやOJTによって向上させてきました。しかし、昨今では「時間がかかりすぎる」、「教える人が足りない」、「育った環境によりバラツキがある」などの問題があり、もっと組織的・人工的・意識的に向上させていく必要が出てきています。

本講座は品質講座の一つで、レビューをテーマとした講座です。本講座の主目的は現場開発のレビュー力を向上させることにあります。欠陥改善の考え方やレビューの進め方と技法・心得を整理復習した後、レビューの助所/レビューの質を向上させるポイント(欠陥除去の視点・原則・経験則)を演習しながら習得していきます。演習することで、実践に適用できる力を身に付けることができます。

加えて講座では、レビューの育成方法なども取り上げます。このことにより、レビューを実践できる環境を整えることができます。

組織のレビュー力の向上には勿論、個人の技術者としてのステップアップにもお役に立ていただければ幸いです。

なお、本講座には、レビューで活用できる「要求・設計・テストに関するエンジニアリング知識」を加えた 2 日版もございます。レビュー対象となる成果物の基本知識もあわせて学びたい方は、ご検討ください。

日 数

1 日

時 間

6.5 時間(10:00～17:30)

受 講 料 (税 込)

550,000 円

※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

ソフトウェア開発者及び管理者

前 提 条 件

特になし

到達 目 標

- レビューの質を向上させるポイントを理解し、開発現場で活用できるようになります。
- レビューの進め方と技法を理解し、開発現場で実践できるようになります。

内 容

- はじめに
- レビューとは
 - 1)レビューの位置づけ
 - 2)レビューの構成
 - 3)レビューの狙い
- レビューの進め方
 - 1)プロセスフロー
 - 2)レビューのルール
 - 3)レビュー会の進め方
- レビューの心得
 - 1)参加者共通の心得
 - 2)モデレータの心得
 - 3)作成者(読み手)の心得
 - 4)レビューアの心得
 - 5)管理者の心得
- レビューの視点
 - 1)フォーカスするレビューの種類
 - 2)レビュー対象の構造
 - 3)レビュー視点の構造
 - 4)共通のレビュー視点
 - 5)仕様レビュー視点
 - 6)設計のレビュー視点
 - 7)チェックリスト
 - 8)演習問題
- よくある問題と対応

※ 演習は、午前中に 1 問、午後には 3 問行います。

【品質】品質分析評価技法講座

— 品質の定量的な把握、捕捉のための基本的な技法を習得する —

"You can't control what you can't measure. -測定できないものはコントロールできない。"というのはソフトウェアエンジニアリングの大家である、トム・デマルコの言葉です。

この言葉には、開発プロジェクトの成功には、開発プロジェクトに影響を与える事象に対していかに迅速かつ適切に対処できるかが重要なのだというメッセージが込められています。

ソフトウェア品質に対しても、品質に影響を与える事象に対する対処が不可欠であり、そのためには適切な品質分析による評価、判断が重要になります。

本講座では、ソフトウェア開発に携わるすべての人を対象に、品質分析・評価に関する重要な概念や活動内容を学んでいただきます。特に品質分析の活動内容(欠陥の特徴づけ、定量的分析、評価・判断)の考え方については、基本的な統計的技法を使った具体例を交えながら紹介いたします。

また、品質分析に特化した尺度を使った分析(ex. 信頼度成長モデル、欠陥密度)については、数式などを学習するのではなく、ソフトウェア開発への適用方法を中心に学んでいただけます。欠陥の分析を中心とした実践的な演習も用意していますので、学んだ内容を即現場にて実践できるようになっております。

日 数

1 日

時 間

7 時間(9:00～17:00)

受 講 料 (税 込)

550,000 円

※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。

※オンライン・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

プロジェクトマネージャ/リーダー、アーキテクト、テストマネージャ、QA マネージャ、テスト設計担当者、開発者、プロセス改善推進者

前 提 条 件

ソフトウェアの開発経験があること

到達 目 標

- 品質の定量的な把握、捕捉の方法を理解する
- 主要な品質分析評価手法を理解し、現場で実践できる(基本的な技法の復習、ソフトウェア開発への適用方法)

内 容

- はじめに
- 品質分析とは
 - 1) 品質分析とは
 - 2) 品質分析の位置づけ
 - 3) 品質分析での実施事項
- 実施事項概要
 - 1) 計量データ収集
 - 2) 特徴づけ
 - 3) 品質計量データ算出
 - 4) 定量的分析
 - 5) 評価・判断
- 代表的な品質分析の手法
 - 1) 欠陥数推移(信頼度成長曲線、信頼度成長モデル、バグ管理図 etc.)
 - 2) 欠陥密度
 - 3) 静的解析との相関関係
 - 4) 欠陥除去効率(DRE)
 - 5) 特徴づけを使った原因特定
 - 6) 特徴づけを使った弱点特定
 - 7) 属性を分析した品質評価・判断
- 品質分析実践例(演習題材)
 - 1) 開発対象の概要
 - 2) T5:プリンタ側統合テスト終了基準
 - 3) T6:システムテストの状況

原因分析ワークショップ

ソフトウェア開発組織の多くで、品質改善の取り組みの一環として不具合の原因分析・再発防止が行われていますが、「難しい」「効果的な再発防止策を出せない」という声もよく聞きます。これにはいくつかの要因が考えられます。人間のミスを取り扱うことにまつわる難しさもあります。過去の出来事について、結果から原因を遡って考えていくのも難しい作業です。根本原因をつきとめたと確信できていなければ、再発防止策にも確信は持てないでしょう。

不具合混入の「よい原因」を突き止め、それに対する「よい防止策」を立案・実行することの重要性は言うまでもありません。本ワークショップは、このように難しいとされる原因分析・再発防止について、その考え方、進め方(プロセス)、注意点を判りやすく示し、具体的な題材を使って実際に体験していただくものです。

座学を中心とした、知識の習得が主目的の講座ではありません。演習での体験を通じて、参加者自身が原因分析・再発防止におけるポイントを感じ、考え、発見していただくことが狙いです。以下のような特徴があります。

題材が、現実起こりがちな不具合を取り上げたものであること、ワークショップ形式の演習により、実際の原因分析に近い体験ができます

体験を振り返ることで、現在の業務の問題点・改善点を考えるきっかけが得られます
演習題材には御社での実例を用いることも可能です(教材のカスタマイズに伴う費用を申し受けます)
ソフトウェアの品質改善に取り組んでいる人全般にお勧めのワークショップです。

日 数

1日

時 間

6時間(10:00~17:00)

受講料(税込)

550,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。
※オンサイト・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。
※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対象者

ソフトウェア開発で原因分析・再発防止の実務を経験したことがある方

前 提 条 件

特になし(ソフトウェア開発、テスト、ソフトウェア品質保証のいずれかを体験しているのが望ましい)

到達目標

- 原因分析・再発防止の考え方、進め方(プロセス)、注意点が判る。
- 不具合発生時の「真の原因」を考え、効果的な再発防止策を見つめることができるようになる。

なぜなぜ分析

問題をロジカルに分析するツールとして話題の「なぜなぜ分析」を使って製造現場で起こっているヒューマンエラーを分析する際の重要ポイントについて解説します。また、グルーワークを行い、分析方法について理解を深めます。

ある問題が発生した場合、その問題を引き起こした要因(なぜ)を考え、さらにその要因(なぜ)を考えることを繰り返すことにより、その問題の真の原因を追究し、その対策として再発防止策を導き出すプロセスを学びます。

本セミナーでは、より上流に目を向けた原因追及を行って、問題の作りこみを防ぐことを目的としています。
つまり「問題を見逃さない」から「問題を作り込まない」現場づくりに視点を変えていきます。

日 数

1日

時 間

6時間(10:00~17:00)

受講料(税込)

550,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。
※オンサイト・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。
※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対象者

- なぜなぜ分析の進め方を理解したいエンジニア・管理者
- お客様に障害および再発防止策の報告をする営業、品質保証部門の担当者

前 提 条 件

- 一般的なソフトウェア開発プロセス(ウォーターフォール開発)の理解
- ソフトウェア開発の経験

到達目標

- 問題が生じた事実を時系列に把握することが出来る。
- 開発プロセスに関する、問題の根本的原因を究明することが出来る。
- 再発防止策として、問題の根本的原因に対する改善提案が出来る。

内 容

形式:講義+演習(個人/グループ)

- はじめに
 - 1) テーマ
 - 2) 目標
 - 3) 進行の目安
 - 4) ルール
- 原因分析・再発防止の概要
 - 1) よく見られる例
 - 2) 原因分析・再発防止の流れとポイント
 - 3) 代表的な手法(参考)
- 演習題材解説
- 原因分析(演習)
 - 1) 個人作業
 - 2) グループ作業
 - 3) 発表・共有
- 再発防止(演習)
 - 1) グループ作業
 - 2) 発表・共有
- 振り返り
 - 1) 振り返り
 - 2) 共有

内 容

形式:講義+ディスカッション+演習

- プロセス改善のためのなぜなぜ分析とは
 - 1) 改善には欠かせないなぜなぜ分析
 - 2) なぜなぜ分析を効果的に行うには
 - 3) プロセスとは
- 問題が発生する要因
 - 1) 問題が発生する二つの要因
 - 2) 開発部門の業務
 - 3) 運用・保守部門の業務
- なぜなぜ分析の進め方
 - 1) なぜなぜ分析の進め方
 - 2) 事実を把握する
 - 3) 問題を作り込んだ要因と流出させた要因に分ける
 - 4) なぜで掘り下げる
 - 5) 根本的原因を見つけ出す
 - 6) 再発防止策を立案する
 - 7) 複数の再発防止策から対策を絞り込む
- グループ演習

講 師

クオリゲート合同会社(弊社パートナー)

※1 本講座はクオリゲート合同会社の著作物です。
※2 (株)豆蔵は本講座の再販売(非独占的)です。



JSTQB 認定テスト技術者 Foundation Levelトレーニング

ソフトウェアテストは、ソフトウェア品質を確保するための重要な手段です。
 やみくもだったり、行き当たりばったりといったテストではなく、ソフトウェアエンジニアリングの基本に則ったテストに組み込まなければ、品質、コスト、納期に見合ったプロダクトやサービスを開発することは困難です。
 しかし実際の現場では、企業や組織間でテストに対する用語一つ取ってみても共通化が行われていないことが多く、テストの取り組み方がばらばらなまま開発が進められていることすら珍しくありません。
 本コースでは、企業や組織の枠を超えたソフトウェアテストの基本的な手法や技法について、国際的に普及している ISTQB のシラバスに基づいて、効率的に学習できます。
 また特徴として、シラバスの内容だけにとどまらず、業界で活躍中のコンサルタントが教育や企業へのコンサル支援の実践の場で得た豊富なノウハウを取り込んだコースとなっています。
 ※ ISTQB はソフトウェアテストの資格認定を行っている国際的な団体です。
 日本では ISTQB が加盟団体となります。詳しくは <http://www.jstqb.jp/> をご覧ください。

日 数

3 日
 ※ ご要望に応じたカスタマイズ(例: 特定部分の深堀、実施日数の調整)も行えますのでご相談ください

時 間

6.5 時間 / 日 (10:00~17:30)

受 講 料 (税 込)

1,650,000円
 ※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。
 ※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。
 ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- 国際的に通用するソフトウェアテストの手法や技法の基礎知識を身につけたい方
- ISTQB-FL レベル資格認定取得に取り組まれている方
- エンタープライズ、組み込みといったドメインに関わらず、開発者、テスト担当者、品質管理担当者、マネージャなど幅広い層が対象です。またユーザ(発注者)の立場として、一般的なソフトウェアテストの知識が必要となっている方も対象となります。

前 提 条 件

- システム開発やソフトウェア開発の経験があると理解が容易になります。(必須ではありません)

例 達 目 標

- ISTQB FL に基づいたソフトウェアテストの基本的な手法や技法を習得する。
- ソフトウェアテストの基礎的な用語を理解し、自身のプロジェクトへ適用できるようにする。

内 容

【第 1 日目】形式:講義 + 演習
 オリエンテーション

1. テストの基礎
 - 1) テストとは何か?
 - 2) なぜテストが必要か?
 - 3) テストの原則
 - 4) テスト活動、テストウェア、そして役割
 - 5) テストに必要なスキルとよい実践例
 2. ソフトウェアライフサイクル全体を通してのテスト
 - 1) コンテキストに応じたソフトウェア開発ライフサイクルでのテスト
 - 2) テストレベルとテストタイプ
 - 3) メンテナンス(保守)テスト
- 初日のまとめ

【第 2 日目】形式:講義 + 演習

3. 静的テスト
 - 1) 静的テストの基本
 - 2) フィードバックとレビュープロセス
 4. テスト分析と設計
 - 1) テスト技法の概要
 - 2) ブラックボックステスト技法
 - 3) ホワイトボックステスト技法
 - 4) 経験ベースのテスト技法
 - 5) コラボレーションベースのテストアプローチ
 - 6) テスト技法の選択
- 2 日目のまとめ

【第 3 日目】形式:講義 + 演習

5. テスト活動のマネジメント
 - 1) テスト計画
 - 2) リスクマネジメント
 - 3) テストモニタリング、テストコントロールとテスト完了
 - 4) 構成管理
 - 5) 欠陥マネジメント
 6. テストツール
 - 1) テストのためのツールによる支援
 - 2) テスト自動化の利点とリスク
- 3 日目のまとめ



C++コーディング・リフレッシュ講座

アセンブラを扱える C 言語との高い上位互換性とオブジェクト指向の特徴を兼ね備えた C++ 言語は今でも利用頻度の高いプログラミング言語の一つです。
 C++ 言語仕様は 1998 年にリリースされた C++98 以降小さな変更しかありませんでしたが、2011 年の C++11 では後発のモダンなプログラミング言語の仕様を取り入れるように大きく変更されました。
 しかもこの大きな仕様変更はこれまでの C++ プログラマには理解し難いものでした。

本講座は、C++11 より前と後のコードを比較しながら C++11 以降の主要な C++ 言語仕様を理解するための講座です。
 またこれら主要な C++ 言語仕様について、導入背景・目的や注意点などプログラミングするにあたって必要な知識を得ることができます。

日 数

1 日

時 間

6 時間 (10:00~17:00)

受 講 料 (税 込)

550,000 円
 ※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
 ※オンライン・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。
 ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- C++11 以降の C++ は、「読みにくい」又は「理解が難しい」と感じている方
- C++03 以前に関する知識と開発経験がある方

前 提 条 件

- C++11 より前の C++ 言語に関する知識と開発経験がある

例 達 目 標

- C++11 以降でどのような言語仕様が追加されたかがわかる
 追加された項目の背景、目的、効果が理解できる
- 追加された項目について、言語仕様の書籍などをしながらプログラミングできるようにする

内 容

形式: 講義+演習(個人/グループ)

1. はじめに
2. C++について
3. Modern C++の哲学
4. 型推論
5. for 文とコンテナ
6. ムーブセマンティクス
7. スマートポインタ
8. オブジェクト
9. ラムダ式
10. 並列プログラミング
11. C++ Core Guidelines
12. おわりに

C言語基礎研修

組込みソフトウェア開発のコーディングの土台となるプログラミング言語「C言語」を基礎から学ぶ研修です。基本文法から関数の作成方法、理解が難しいとされている、ポインタや構造体、共用体メモリを重点的に、講義と演習を通じて系統的に順序だてて学習します。

演習では、キーボードやディスプレイなどの標準入出力の操作方法から、効率的なデータ処理方法、ダブルポインタやポインタ配列、さらに動的メモリ領域の取り扱いなどについて学習します。

cygwinの環境を使用しC言語プログラムを自力で書くことができることを目的とした研修です

日 数

10日

時 間

7時間/日(9:00~17:00)

受講料(税込)

550,000円~5,500,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※オンサイト・オンライン(Zoom等のオンライン会議システムを利用して実施可能)ともに同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

特に制限なし

前 提 条 件

特になし

到達目標

- C言語の基礎的なプログラム技術を得得する。
- 理解度確認テストを実施し、8割以上の点数を獲得。

内 容

【講義項目】

- 第1章 コンピュータ基礎
- 第2章 C言語基礎知識
- 第3章 変数
- 第4章 文字
- 第5章 演算子(1)
- 第6章 演算子(2)
- 第7章 制御文(選択)
- 第8章 制御文(ループ)
- 第9章 制御文(その他)
- 第10章 関数
- 第11章 メモリモデル
- 第12章 変数スコープ
- 第13章 関数スコープ
- 第14章 配列
- 第15章 ポインタ(1)
- 第16章 ポインタ(2)
- 第17章 アクティビティ図
- 第18章 構造化プログラミング
- 第19章 構造体
- 第20章 アライメント
- 第21章 共用体
- 第22章 エンディアン
- 第23章 文字列
- 第24章 メモリアロケート/フリー
- 第25章 関数ポインタ
- 第26章 プリプロセッサ

- 第27章 typedef宣言
- 第28章 enum宣言
- 第29章 修飾子
- 第30章 Makefile
- 第31章 組み込み基礎知識
- 第32章 高度なCテクニック

【演習項目】

- 課題01 キーボード入力と合計加算
- 課題02 引数のポインタ渡し
- 課題03 演算子
- 課題04 ビットフィールド
- 課題05 ビットシフト
- 課題06 関数名ポインタ
- 課題07 ファイルIO
- 課題08 分割コンパイル
- 課題09 棒グラフの表示
- 課題10 構造体のソートと表示
- 課題11 ファイルの読み込み 単語数カウンタ
- 課題11_参考課題A ファイルダンプ
- 課題11_参考課題B ファイルダンプ(オプション指定)
- 課題12 チェイン構造によるキュー管理
- 課題12_参考課題A チェイン構造によるキュー管理 (UNION仕様)
- 課題12_参考課題B チェイン構造によるキュー管理 (ハブルソート)

ワークショップ(ライトレース+衝突回避)

STEP1: エンジニアリング基礎、STEP2: 手法・技法系講座で学んだ手法を生かし、仮プロジェクトを実際に運用しながら組込み開発を一通り体験いただく講座です。(要件整理、設計、実装、検査等を一通り体験いただけます)。

4~5名程のGrでプロジェクトを進めていき、その中で進捗MTGや成果物レビュー、振り返り等、プロジェクト開発には欠かせない工程も実践的に学ぶことができます。

■本講座で扱うライトレースカーの機能:

- ①赤外線センサを用い、決められた線上を自動走行する
- ②超音波センサを用い、衝突回避を行う

日 数

5日

時 間

7時間/日(9:00~17:00)

受講料(税込)

550,000円~2,750,000円

※オンサイトでの開催となります。

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- プロジェクトメンバーとして開発に携わる経験を積みたい方
- 座学で学んだ知識を、実践で活用したい方

前 提 条 件

- 要求、設計、実装、テストに関する基礎知識を有する方(学んだことがある方)
- UMLの基本知識(クラス図、シーケンス図、ステートマシン図、パッケージ図、コンポーネント図、パッケージ図、配置図)を有する方(学んだことがある方)
- C言語の基礎的なプログラミング技術を有する方(学んだことがある方)

到達目標

- プロジェクト開発の流れを理解することができる
- チームの中での立ち回り方や、各種設計技法の理解を深めることができる

内 容

形式:講義 + 演習

1. はじめに
2. ライトレースカーについて
 - 1) ライトレースカーの説明
 - 2) 搭載されている機能の説明
 - 2.1) 赤外線センサ
 - 2.2) 超音波センサ
3. 開発環境とコーディングについて
 - 1) 使用環境(Arduino IDE)の紹介
 - 2) Arduino IDEにおけるコードの書き方
4. 演習
 - 1) 赤外線センサを用いたライトレース(1週目)
 - 2) 超音波センサを用いた衝突回避(2週目)



ワークショップ(イメージスティック)

弊社オリジナル製品「イメージスティック」を用いた組み込み開発の体験講座です。

■イメージスティックの紹介

16個のLEDとタクトスイッチ、Gセンサが取り付けられた棒状の組み込み機器、スイッチ入力やGセンサ値を取得して、LEDの点灯パターンを切り替える課題を実施します。

「ワークショップ(ライトレース+衝突回避)」同様に、チームプロジェクト形式で要件整理から検査までの工程を一通り体験いただけます。

※タイマ割り込み等を用いるため、難易度はより高くなります。

日 数

10日

時 間

7時間/日(9:00~17:00)

受講料(税込)

550,000円~5,500,000円

※オンサイトで開催となります。

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- プロジェクトメンバーとして開発に携わる経験を積みたい方
- 座学で学んだ知識を、実践で活用したい方

前 提 条 件

- 要求、設計、実装、テストに関する基礎知識を有する方(学んだことがある方)
- UMLの基本知識(クラス図、シーケンス図、ステートマシン図、パッケージ図、コンポーネント図、パッケージ図、配置図)を有する方(学んだことがある方)
- C言語の基礎的なプログラミング技術を有する方(学んだことがある方)

到達目標

- プロジェクト開発の流れを理解することができる
- チームの中での立ち回り方や、各種設計技法の理解を深めることができる

内 容

形式:講義 + 演習

- はじめに
- イメージスティックについて
 - イメージスティックの説明
 - 搭載されている機能の説明
 - LED(赤、青、緑の三色点灯)
 - タクトスイッチ
 - Gセンサ
- 開発環境とコーディングについて
 - 使用環境(MPLAB@X IDE)の紹介
 - MPLAB@X IDEにおけるコードの書き方
- 各種センサ類の操作方法説明
- 演習
 - LEDを各色点灯させる
 - タイマ割り込みを用いたLED切り替え
 - スイッチを用いたLED切り替え
 - スイッチの単押しと長押し
 - スイッチ入力時間を用いた制御
 - グラデーション表示
 - グラデーションの移動
 - パターン描画

※全課題毎に要件整理フェーズを含みます。



(目的別セット講座)UMLによる上流設計基礎講座

企業の開発現場では、これまで個々人の経験やスキルに依存した開発が行われてきました。しかし現在は、ソフトウェアが製品価値の中心を担う時代となり、体系的な知識に基づく開発風土を社内に根付かせることが急務となっています。

本講座セットは3日間でUMLによる上流設計の基礎力を養成し、エンジニアリングに基づく開発の考え方やUMLを活用する風土を醸成することを目的とした集中プログラムです。単独講座である「エンジニアリング基礎講座」「UML入門講座」「UMLモデリング演習講座」の3つを組み合わせ、システム開発の上流工程で作られる「概念モデル」や「分析モデル」といった抽象度のモデルを描き始めるために必要な基礎知識や知見を効率的に学習できるよう比重を構成しています。

プログラムでは、エンジニアリング活動の全体像を押さえ、UMLをどの工程で活用すべきかを理解します。そのうえで、曖昧になりがちな表記法を体系的に学習し、さらに、多数の演習問題を通じて、表現の工夫や失敗から学ぶモデリングの勘所を体得、現場で自ら使い始められる実力を養成します。

豆蔵のエンジニアリング教育は、単なる表記やプロセス紹介ではなく、「何を解決すべきか」「システムが対象としている世界をどう捉えるか」を重視しています。本講座ではその強みを活かし、実務課題に直結するUMLモデリング能力を引き上げます。

これまで数千人が受講し、継続導入いただいている企業様からは「開発プロセスの改善や組織的な開発力強化につながった」と高い評価をいただいています。わずか3日間で、UMLモデリングを業務に適用できる力を獲得できる講座です。

日 数

3日

時 間

7時間(9:00~17:00)

受講料(税込)

1,650,000円

※想定受講者数は20名です。受講者数が20名を超える場合は費用が変わります。

※オンサイト・オンライン(Zoom等のオンライン会議システムを利用して実施可能)とも同料金です。

※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- ソフトウェア開発に携わる人
- UMLモデリングを始めたいと考えている人

前 提 条 件

- オブジェクト指向の基礎概念を理解していること
- 何らかのプログラミング言語での実装経験があること

到達目標

- 自分が所属する組織のメンバーにエンジニアリングの中で行われる活動の実施事項とUMLモデルの使われ方を説明できる。
- UMLとは何か、採用のメリットを人に説明できるようになる。
- UMLで記述されたモデルを理解できるようになる。
- UMLのいくつかの図を現場で使い始められるようになる。
- モデルによって伝達したいことを表現できるようになる。
- オブジェクト指向モデリングの要領・勘所を習得し、通常のレベルの問題であれば適切にモデリングできるようになる。

内 容

【1日目】

午前

プロセス編(座学)：典型的な開発プロセスの中でUML図がどのように使われるのか概要を把握する

午後

表記法編(座学+演習)：UML表記法のおもな要素の描き方/解釈の仕方を学習する

【2日目】

午前

表記法編の続き

午後

モデリング演習編(演習中心)：習得した表記法を使って、いくつかの小さな題材で実際にモデルを描くことで理解/知見を深める。

【3日目】

午前

モデリング演習編の続き

午後

モデリング演習編の続き

理解度確認テスト(オプション)

【目的別セット講座】要求開発・要件定義連結講座

ー 超上流工程からシステム要件定義を組織で一貫して学習する ー

企業のプロダクト開発においては、経営層の事業戦略と現場の開発活動が十分に結びつかず、現場の判断や想いだけで進めた結果、投資回収がでない製品や、予算停止に至るプロジェクトが少なくありません。

本来、プロダクトやサービスは企業戦略の実現手段であり、経営層の意図から利用者の課題、そして開発対象の要件へと一貫して落とし込むことが不可欠です。

本講座セットは、事業戦略と現場の開発を一貫して結びつけ、要求から要件定義までを組織として確実に落とし込む力を養成することを目的とした集中プログラムです。単独講座である「要求開発実践講座」「UML入門講座」「要求基礎講座」の3つを組み合わせ、効率的に学習できるように構成しています。

要求開発実践講座で学ぶ「事業戦略から要求を導くプロセス」を出発点に、UML入門講座で要求を要件に翻訳するための表現力を習得し、要求基礎講座で要求を要件定義へつづく手法を体系的に学びます。従来は個別に受講していた内容を一連の流れに体系化することで、事業戦略から開発現場を一本の筋で結び、事業戦略を具体的なプロダクト開発へと結びつける力を養成します。

単なる知識習得にとどまらず、組織として事業目標を確実にプロダクト要件へと落とし込む体制を築きたい企業に最適な講座セットです。

日 数

要求開発編(受講生 A)： 3 日
UML 表記法編 + 要件定義編(受講生 B)： 2 日

時 間

要求開発編(受講生 A)： 7 時間/日(9:00~17:00)
UML 表記法編 + 要件定義編(受講生 B)： 1 日目 7.5 時間(9:00~17:30)、2 日目 7 時間(9:00~17:00)

受 講 料 (税 込)

1,100,000 円 ~1,650,000 円
※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
※オンライン・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)ともに同料金です。
※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- プロダクトやサービスの企画に関わる立場の方(要求を出す側および要求から企画する側とも) → 受講生 A
- 企画に基づいて、要求獲得・要求分析・プロダクトやサービスの要件作成を行う立場の方 → 受講生 B

前 提 条 件

- 要求開発編(受講生 A)
 - 組織におけるプロダクト/サービスの企画や開発に関わった経験が5年以上あること
- UML 表記法編 + 要件定義編(受講生 B)
 - システム開発の経験が 3 年以上あり、要求者からの要求獲得の経験があること

到 達 目 標

- 要求開発編(受講生 A)
 - 経営層の事業戦略、利用者の課題・要求、プロダクトへの要求や制約を整理し、事業戦略からプロダクトの要件への一貫性を作り出す方法を人に説明できる

オブジェクト指向分析設計講座

ー マネージャ向けー

組込みシステム開発へオブジェクト指向を適用するという話は様々なところで話題になっています。しかし、オブジェクト指向を導入すると何がかわるのか、どんなものを作るのか、何が難しいのかといったことの具体的な部分をマネジメントの視点から説明した情報や、そういったことを取り上げた講座は世の中にあまりありません。本講座はそうした組込みシステム開発へのオブジェクト指向適用をマネジメントの観点から取り上げた、非常にユニークな講座となっています。

本講座は、プロジェクトマネージャやチームリーダーといった開発をリーディングする立場の方々オブジェクト指向開発の進め方とマネジメント上のポイントを習得するための講座です。

オブジェクト指向を開発に導入することで、ソフトウェア開発とマネジメントがどのように変わるのか/変わらないのか、ソフトウェア開発としてどのような活動を行い、どういった成果物を作るのか、また、マネジメントとしてそれらの成果物をどうチェックしたら良いのかといったことを、具体的な開発工程(要求定義~要求分析~設計)を追いながら学んでいくことができます。

また、この講座を受講することで一般的なマネジメントやプロセスとオブジェクト指向がどのように対応するのかも学ぶことができます。

オブジェクト指向の導入を考えている管理者の方、導入はしたけどマネジメントの仕方が良く分からなくて困っている方に最適な講座ですので、オブジェクト指向絡みの悩みを抱えている管理者の方は是非受講してみてください。

日 数

1 日

時 間

6 時間(10:00~17:00)

受 講 料 (税 込)

550,000 円
※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
※オンライン・オンライン(Zoom 等のオンライン会議システムを利用して実施可能)ともに同料金です。
※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

- 組込みシステム開発におけるオブジェクト指向の適用方法を知りたい方
- オブジェクト指向開発におけるプロジェクトマネジメントについて知りたい方
- UML の組込みシステム開発向けの使い方を知りたい方

前 提 条 件

- オブジェクト指向の基礎用語(クラス、インスタンス、継承、カプセル化など)を理解している。
 - ※ 弊社トレーニング「わかるオブジェクト指向」修了程度の知識
- UML 表記法の基礎を理解している。
 - ※ 弊社トレーニング「UML によるモデリング入門」修了程度の知識
- ソフトウェア開発の経験がある。
- ソフトウェア開発のプロジェクトマネジメント経験がある。

到 達 目 標

- オブジェクト指向を適用した開発における活動(要求定義、要求分析、設計)の目的を説明できる。
- オブジェクト指向を導入することで、変わる部分と変わらない部分を説明できる。
- オブジェクト指向開発での各作業成果物について技術的に確認すべき点とマネジメントでの利用法について説明できる。

- 利用者とプロダクトの関係を図で可視化し、利用者の課題と解決策の検討方法を人に説明できる
- 利用者に価値をもたらすユースケースを立案する流れを人に説明できる

■ UML 表記法編 + 要件定義編(受講生 B)

- UML で表現された要求開発の成果物(業務フローや概念モデル)を正しく読み解き、それらを洗練して関係者と合意形成することができる
- 要求元を明確化し、要求を獲得するプロセスを人に説明できる
- 要求開発で設定したゴールを、システムの要求へつなぎ、要求分析から要件定義に落とし込む方法を人に説明できる

内 容

要求開発編

形式: 講義+演習

【第 1 日目】

I 要求開発実践講座の概要

1. 要件定義の基礎知識
2. 要求開発の要旨
3. システム開発を前提にした要求開発

II ゴールモデリングの基礎講座

1. モデリングの基礎知識
2. ゴールモデリングの基礎知識
 - 2.1 ゴールモデリングの位置付け
 - 2.2 ゴールモデリングワークショップ

【第 2 日目】

II ゴールモデリングの基礎講座 (続き)

- 2.2 ゴールモデリングワークショップ(続き)

III ビジネスモデリングの基礎講座

1. ビジネスモデリングの基礎知識
 - 1.1 ビジネスモデリング
 - 1.2 サービスモデリング
 - 1.3 プロセスモデリング

【第 3 日目】

III ビジネスモデリングの基礎講座(続き)

- 1.3 プロセスモデリング(続き)
- 1.4 情報モデリング
- 1.5 あるべき業務モデルの設計

IV システム開発へのつながり

1. システム開発へのつながり
2. まとめ(振り返り)

UML 表記法編 + 要件定義編

【1 日目】

午前

- UML 表記法編(1)： 要求開発の成果物とUMLダイアグラムの対応を理解する
- UML 表記法編(2)： UML 表記法のおもな要素の描き方/解釈の仕方を学習する

午後

UML 表記法編の続き

- 要求定義編(1)： 要求開発の成果物と要件定義の対応を理解する

【2 日目】

午前・午後

- 要求定義編(2)： 要求開発の成果物を踏まえつつ、要求獲得・要求分析・要件の作成の仕方を学習する

内 容

1. はじめに

2. 開発全体の構造

- 1) これまでの開発手順のおさらい
- 2) オブジェクト指向開発で変わる点
- 3) オブジェクト指向開発の特徴
- 4) オブジェクト指向開発の全体像

3. 要求定義

- 1) 要求定義の目的
- 2) 要求の獲得からシステム要件作成まで
- 3) 要求定義の手順と成果物
- 4) 従来開発との違い
- 5) UseCase の効果
- 6) マネジメントの着眼点

4. 要求分析

- 1) 分析の目的
- 2) オブジェクト指向での重要な考え方
- 3) オブジェクト指向での構造の捉え方
- 4) 分析の手順と成果物
- 5) シナリオ分析
- 6) オブジェクト指向分析
- 7) 従来開発との違い
- 8) オブジェクト指向分析の効果
- 9) マネジメントの着眼点

5. 設計

- 1) オブジェクト指向設計の考え方
- 2) オブジェクト指向設計の概要
- 3) オブジェクト指向設計の進め方
- 4) アーキテクチャ
- 5) オブジェクト指向設計の一般的な詳細手順
- 6) 分析と設計の違い
- 7) 従来開発との違い
- 8) オブジェクト指向導入による効果
- 9) マネジメントの着眼点



発注者向け 要件定義講座

近年、多くの企業が自社のサービスや製品に付加価値を与えるためにソフトウェア開発を必要としています。しかし、発注者側が持っているのは自社の事業やサービスに関する知識(ドメイン知識)であり、必ずしもソフトウェア開発の知識や経験ではありません。そのため、外部の開発企業にソフトウェアの開発を発注する際に、伝えるべき要求が十分に整理されていなかったり、曖昧なまま契約に進んでしまったりすることが少なくありません。その結果として、完成したシステムが期待とは異なるものになり、修正や追加のために多大なコストや時間がかかるという問題が繰り返されています。

特に社内にソフトウェア開発者がほとんどいない企業では、ソフトウェア開発の知識のない事業部門の社員が、要求をまとめ、発注者として開発会社に説明を行い、合意形成まで行わなければならないことは、避けて通れない課題となっています。

本講座では、発注者として必要な要求定義に関する基礎知識を習得し、発注や受入れに伴うトラブルを未然に防ぐ力を養います。実際のシステム開発でよく起こる問題を題材にしながら、要求の獲得、文書化、合意形成といったプロセスを体系的に理解することができます。また、講義だけでなく演習を通じて、現状の業務を整理し記述する方法や、条件の組合せを明確化する技法、業務記述のレビューを通じた改善の視点、シナリオ分析を用いた具体的な要求抽出、さらには要求の精査といった実践的な取り組みを体験します。これにより、単なる知識の習得にとどまらず、実際に直結するスキルを身につけることが可能となります。

自社の事業を深く理解している発注者が、この知識と経験を身につけて開発側と協働することで、システムは単なる外注物ではなく、事業成長を支える真の基盤へと進化します。本講座の最大のメリットは、発注者自身が主体的に関与する姿勢を養える点にあり、ベンダーに任せきりにせず、自社に必要な機能や品質を正しく言語化し共有する力を持つことが、プロジェクト成功への確実な第一歩となるのです。

日 数

2 日

時 間

4 時間 / 日 (13:00~17:00)

受講料 (税込)

660,000 円

- ※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
- ※オンライン・オンライン (Zoom 等のオンライン会議システムを利用して実施可能) ともに同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

ソフトウェア開発の発注を行っている人、これから行う人

前 提 条 件

特に無いですが、発注の経験があるとより聞きやすいです

到達目標

- 以下の重要な項目をよく理解し、人に説明できるようになる
 - 要求の種類、特徴
 - 要求の精度を高めることの重要性
 - 要求の精度を高めるための方法
- 要求や要件に関する活動について、自組織の弱いところを発見し、改善の糸口をつかむ

内 容

- はじめに
- 要求の重要性と問題
 - 2.1 エラーの主要原因
 - 2.2 よく目にする好ましくない現象
 - 2.3 要求・要件・仕様的一般に認知されている問題点
- ソフトウェア開発の特徴
 - 3.1 ソフトウェア開発とはどんなものか
 - 3.2 ソフトウェア開発に関して知っておいてほしいこと
- 要求とは？
 - 4.1 要求とは
 - 4.2 言葉の定義
 - 4.3 要求の特徴
- 要求に関する開発活動の概要
 - 5.1 プロジェクト立ち上げ時の活動 ~ 先立つ活動~
 - 5.2 要求を要件化するための活動
- 要求の獲得
 - 6.1 要求獲得のステップ
 - 6.2 要求元の特定
 - 6.3 要求の抽出
 - 演習 1: 業務の記述
 - 演習 2: ディビジョンテーブルの活用
 - 演習 3: 業務記述のレビュー
 - 演習 4: シナリオを使った分析
 - 6.4 要求の文書化
 - 演習 5: 要求の精査
- 開発側との合意
 - 7.1 要求の合意
 - 7.2 要件の合意
 - 7.3 合意内容の変更
- まとめ



発注者向け 受入れテスト講座

システム開発を外部に委託している企業は数多くありますが、システム納品時の確認は十分にできているでしょうか？ 契約通りの納品物が揃っているかの確認だけで内容の確認がおざなりになってしまっていないでしょうか。実際の業務で問題なく利用できるか、非機能要件も含めて期待通りに動作するかを確かめる「受入テスト」が不十分であることが原因で、運用開始後に深刻なトラブルの発生や、改修のための追加コストの発生に頭を悩ませている企業は後を絶ちません。

その一方で、発注者側の担当者は必ずしもテストの専門知識を持っているわけではありません。そのため、受入テストの計画や進め方を十分に理解しないまま検収作業に追われ、契約期間が過ぎてから不具合に気づき、改修に多大な費用がかかるといった事態に直面することも少なくありません。こうした状況を防ぐためには、発注者自身が受入テストの基本を理解し、計画立案から実施、課題への対処に至るまでの一連のプロセスを主体的に管理できる力を身につけることが重要です。

本研修では、受入テストに関する基本的な考え方や用語を学び、良いテストを実現するために必要なプロセスを体系的に理解します。システムテストと受入テストの違いを踏まえ、計画書や仕様書の作成方法を実践的に学びます。さらに、テストの実施や課題への対処、終了判定の考え方についても習得します。講義に加えて演習を行い、業務シナリオを基にしたテストケースの準備作業を体験することで、知識を実務に直結するスキルへと高めることができます。

研修を通じて、自組織の受入テストにおける弱点を発見し、改善の手がかりを得ることができるようになります。外部委託が一般化する現代において、発注者が主体的に品質確認に関与することは不可欠です。本研修は、システムを円滑に業務へ適用し、安心して活用するために必要な知識と実践力を養う場となります。

日 数

1 日

時 間

6 時間 (10:00~17:00)

受講料 (税込)

550,000 円

- ※想定受講者数は 20 名です。受講者数が 20 名を超える場合は費用が変わります。
- ※オンライン・オンライン (Zoom 等のオンライン会議システムを利用して実施可能) ともに同料金です。
- ※受講料は予告なく変更・改定する場合がございますので、あらかじめご了承ください。

対 象 者

今後、情報システムの調達に際し、受入テスト、もしくは、その管理をする人

前 提 条 件

- 受講のための前提条件は特にありません
 - 調達、及び、受入テストの経験がある方は自信の経験と照らし合わせる形で聞いて頂くより理解しやすくなります

到達目標

- 受入れテストに関連する基本的な考え方や用語を理解する
- 良いテストを実現するためのテストプロセスの必要性と内容を理解する
- 発注側で実施する開発管理について必要性と内容を理解する
- 受入れテストについての理解を深め、自組織で行っている活動の弱いところを発見し、改善の糸口をつかむ

内 容

- はじめに
- テスト業務の全体像
 - 2.1 受入テスト未実施、不完全だった場合に想定される影響
 - 2.2 受入れテストの目的
 - 2.3 システムライフサイクルにおけるテスト業務の全体像
 - 2.4 V 字モデルにおける受入テストの位置づけ
 - 2.5 テストに関する用語
 - 2.6 テストプロセス
- 【参考】テストケースにたどり着くまでのステップ
- システムテストと受入テストの違い
 - 3.1 テストレベル
 - 3.2 システムテストと受入テストの具体的な違い
 - 3.3 システムテストの不備、受入テストの不備による影響
- 受入テスト計画書の作成
 - 4.1 テスト計画とは
 - 4.2 テスト計画で検討すべき項目
 - 4.3 受入テスト計画書の作成
- 受入テスト仕様書の作成
 - 5.1 テストケースの作成
 - 5.2 要件項目とテストケースの関係
 - 5.3 テストケース作成時の留意事項
- 【参考】組合せ数による故障検出率
- 5.4 受入テスト仕様書の主要な記載項目と記載内容
- 演習 1 テストケースの作成
- 受入テストの実施、課題検出時の対処
 - 6.1 受入テストの実施と報告
 - 6.2 検出された課題の管理と対処
 - 6.3 受入テストの終了判定
- 受入テスト実施における留意点
 - 7.1 受入テスト環境における留意事項
 - 7.2 テストケースの管理における留意事項
 - 7.3 受入テストデータにおける留意事項
 - 7.4 開発中、及び、受入テスト前の委託先の品質活動の確認