

Light Weight な UCD 手法の提案
Scrum に UCD 手法は適用できるのか？
Proposal of Light Weight UCD method
Can we adapt UCD methods to Scrum process or not?

主査		金山 豊浩	HCD-Net
副主査		福山 朋子	(株)インテック
アドバイザー		篠原 稔和	ソシオメディア(株)
研究者	リーダー	小渕 一幸	セイコーエプソン(株)
		南齋 雄一	(株)アドバンテスト
		小澤 純	(株)山武
		小林 誠	USOL 東京(株)
		清水 覚	伊藤忠テクノソリューションズ(株)
		林 恵子	伊藤忠テクノソリューションズ(株)
		田上 貴久	アンリツエンジニアリング(株)
		村上 和治	東京海上日動システムズ(株)

研究概要

従来から広く用いられているウォーターフォールのような直列型のプロセスは、下流工程での手戻り作業の損失が大きいため、想定される損失規模に見合う UCD 手法導入費用を許容できた。この場合、上流工程での的確かつ精度の高いニーズを把握することが UCD 手法で重要になる。

しかしながら、UCDの本質は、ユーザビリティ向上、すなわち、「ある製品が、指定された利用者によって、指定された利用の状況下で、指定された目的を達成する際の、有効さ、効率及び利用者の満足度の度合い(付録 2)」を高めることにある。したがって、UCD手法は、繰り返し型の開発プロセス（例えば、アジャイル開発プロセス）においても有効に活用できると思われる。

本研究では、アジャイル開発プロセスの一つである Scrum を題材に、各 UCD 手法が適用可能な場所へマッピングし、その中で特に効果のありそうな場所を抽出した。また、より手軽に、素早く UCD 手法が利用可能なように、シンプルな UCD 手法のプロセスを定義した。これにより、繰り返し型のプロセスにおいても UCD 手法の有効な活用を行うことができると考える。

Abstract

The sequential development process widely used such as the Waterfall development process of which efforts of re-working is high correspond to cost and time of an introduction of UCD methods. In this case, it is important that stakeholders understand the user needs with high adequacy and accuracy at upper phase.

However, the essence of UCD is improvement of the usability, that is, improvement of "The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments". Therefore, we think that we can use the UCD methods effectively in the iterative and incremental development process at any phase.

In this study, we selected Scrum that was one of the agile development processes as a study theme. We mapped each UCD methods to Scrum process, and have extracted effective processes for using UCD methods. Moreover, to use them more easily, and quickly, we defined simple UCD method processes. As a result, we think that we can use the UCD methods effectively in the iterative and incremental development process.

1 はじめに

1.1 テーマ選定と背景

従来の UCD (User-Centered Design, ユーザー中心設計) 手法は、ユーザビリティの向上を目的とする要件を「目標の不確実性」と捉え、開発初期段階において大きく「目標の不確実性」を減らすアプローチをとってきた。これは開発上流工程に要件定義の精度を上げるウォーターフォール的な開発プロセスを考えた場合には導入効果が期待できた。しかし、ユーザーが開発初期段階において確かな要求を持っているとは限らない。ユーザーの要求は開発途中で変わる可能性があるため、これは非常にリスクの高い行為であると考えられる。

昨今、ソフトウェア開発プロジェクトにおいてユーザーを中心に捉え設計を行うには、短いサイクルでユーザーからのフィードバックをもらい「目標の不確実性」を徐々に減らしていく開発プロセス(XP、Scrum に代表されるアジャイル開発プロセスなど)が注目されている。これはウォーターフォール的な開発プロセスと比較した場合、ユーザーを中心とした開発がより実践的に行えるためである。しかし、こういった開発プロセスにおいて現状の UCD 手法は、実践するにはあまりにも重いプロセスである。そのため実際のプロジェクトにおいて実践可能な軽い (Light Weight な) UCD 手法が必要であると考えた。

1.2 本年度の活動目標

アジャイル開発プロセスの一つである Scrum において、シナリオ、ペーパープロトタイピング、ユーザビリティ評価といった UCD 手法を効果的に実践できる活用方法を明らかにする。活用に向け、Light Weight な UCD 手法を定義し直し、Scrum 内において実践するための提案を行う。

1.3 研究の進め方

研究活動を行うにあたり、アジャイル開発を体系的に理解しているメンバーが少なかったため、まず手法の習得から開始し、習得した手法を元に UCD を軽量化するためのプロセス構築を行った。具体的には以下の手順で研究を進めた。

(1) アジャイル開発手法の習得

- 外部より講師を招聘し、事例研究を実施する (3回)。
- 演習コースと合同でアジャイル開発勉強会を実施する。

(2) UCD手法の体得

- 代表的な UCD 手法が、軽量な開発プロセスの中でうまく機能するかどうかを検討する。
- UCD 手法を仮想のソフトウェア開発に適用してみることで、その特性や有用性を把握する。

(3) UCDプロセスの仮説立案

- Light Weight な UCD プロセスを定義する。

(4) 仮説の洗練

- KPT (KEEP/PROBLEM/TRY) による振り返りを行い、定義したプロセスの実現可能性を検討する。
- 検討結果からプロセスを再定義し、さらに検討を繰り返すことで、仮説を洗練させる。

2 対象とする手法の解説

2.1 開発手法

近年ソフトウェア開発において、計画時点で明確に要件定義を行うことが難しいケースや計画時点で決めたことが計画を進めるうちに大きく変化し、ウォーターフォール型開発手法だけではうまく対処できないケースも認識された。そのウォーターフォール型開発手法への反省として、アジャイル開発手法が検討されてきた。

ウォーターフォール型開発手法では、計画の詳細化とその順守によってソフトウェアの品質を高めようとするのに比べ、アジャイル開発では、その意味のとおり、素早く変化へ適合することによって、ソフトウェアの品質を高めようとしている（アジャイル開発の定義については付録1を参照）。

アジャイル開発手法の特徴としては、変化に対する受容とユーザーと開発者との相互連携の強化という点がある。（アジャイルソフトウェア開発マニフェストについては付録1を参照）

1990年代後半より、様々なアジャイル開発手法が提案されてきた。XP（エクストリームプログラミング）やScrum、Crystal（クリスタル）、及びリーンソフトウェア開発（LSD）等はそれらの一つである。

Scrumとは、現在多くの開発者の賛同を得つつあり、シンプルな開発プロセスを持つアジャイル開発手法の一つである。Scrumの特徴としては、スクラムマスターというファシリティ管理者をもち、プロダクトの責任者であるプロダクトオーナーと開発者であるチームメンバーが自立的に立ち回り、短い間隔での開発（スプリント）とフィードバックを繰り返していく点がある（Scrumの用語は付録1.1を参照）。

2.2 UCD手法

UCDとは、例えばプロトタイピングなど、実際に動くものを作成してユーザーにユーザビリティ（使い勝手）を検証してもらう手法である（ISOによるユーザビリティの定義については付録2を参照）。

今回、アジャイル開発との親和性を検討するUCD手法[1]には以下の4つである。

(1) シナリオ

シナリオとは、ユーザーがどのような状況で何をするかという、ユーザーに視点を置いた文脈とタスクに関する情報が表現されたシナリオをもとに、ユーザーの要求を分析し、システムの問題点の検討等を行う手法である（文献[1]を参照）。

(2) ペーパープロトタイピング

ペーパープロトタイピングとは対象システムの画面のプロトタイプを低コスト、短時間で作成できる「紙」で作成し、ユーザビリティを評価する。

(3) ユーザーテスト

ユーザビリティテストとは、実際のユーザー（もしくはユーザーを模した被験者）が予め決められたテスト項目を実行する状況を観察し、ユーザビリティを評価する。文献[1]参照。

(4) ヒューリスティック評価

ヒューリスティック評価とは、製品、またはその仕様書やプロトタイプを検査して、そのインターフェースに問題がないかどうかを評価する手法である。評価者はガイドラインに基づいて経験則によって問題を発見する。文献[1]を参照。

3 UCD手法の改善

3.1 実現可能性検討

アジャイル開発に UCD 手法を用いた実践を行い、繰り返し型のプロセスにおいても有効的にかつ実現可能であるのかを実践し検討を行った。

3.1.1 作業計画

UCD 手法の実践検証にあたり、まず全体の作業計画を明確にした。

(1) UCD手法に適合している開発プログラム（題材）を選定する

⇒サンプル「工数登録システム」に決定。

【工数管理表と関連作業の流れ】

1. 部署内での案件コードをもとに日付・時間単位で作業タスクを「工数登録」（EXCEL ファイル）に入力し、「CSV 出力」ボタンで CSV ファイルを出力し社内システムへアップロード。
2. 内容を確認後、印刷。
3. 工数登録画面と同じ内容を、毎日メールで手動入力して上司に報告。

【システムの問題点とユーザー要件】

- 工数登録～日報報告までのオペレーションが多すぎる。
- もっとオペレーションをシンプルにしたい。
- 工数登録画面については付録 3 「工数登録システム」を参照。

(2) 開発プログラム（題材）を元に短い時間の中で繰り返し型の開発を実践する

- インタビューを行う。ユーザーの一日の業務内容をヒヤリングし、システムの目的とユーザー要件および条件を確認する。
- シナリオとペーパープロトタイピングを作成する。
- 作成したシナリオとペーパープロトタイピングを元にユーザレビューを行う。
- 改善案を複数パターンで提示する。ユーザーの要望リストを作成する。
- 次のサイクルでの作業範囲を決め、「改善 ～ 評価」を繰り返してみる。

(3) 開発プログラム（題材）を短時間でUCDの観点で開発するために、UCD手法をどのように改善したいのか、検討する

上記の作業をユーザーと開発者 2 チームに分かれ（仮想のシステム開発の）実践を行った。

3.1.2 検討結果

○ユーザーの業務行動を理解し、ユーザー自身も気づけなかった改善案を短期間で出すことができた。

×インタビュー時間が不足し、記録が分かりづらくなった。

○シナリオにユーザー要件を明確にまとめることで、チーム内で情報共有することができた。

○シナリオとペーパープロトタイピングとペアで行って見たところ、具体的なイメージが倍増してユーザーに伝わった。

○1 サイクルごとにシナリオを作成して改善案を提示することにより、短時間でユーザレビュ

ーまで実施することができた。

○ユーザーも確認ポイントを絞って短時間で評価を出すことができた。

○短期間にレビューすることで、ユーザーとずれている内容については次回の課題としてすぐに取り入れることができた。

○1 サイクルごとにユーザーを含めて振り返りを実施でき、開発における悪い点、良い点などの改善策もより早く見出すことができた。

×シナリオベースの説明だけでは利便性がユーザーに伝わり難いと思われた。

上記の検討結果から、アジャイル開発（繰り返し型のプロセス）においてもUCD手法を用いることは有効で、実現可能であるという見込みができた。一方、以下の点で、そのままUCD手法を実施することは難しいことがわかった。

- 短時間では中途半端な実践になる
- 手法に向き不向きがある
- 人により手順にばらつきがでる

今回の実現可能性検討を念頭に、代表的なアジャイル開発手法の一つであるScrumにおけるUCD手法を効果的に実用するための手法を3.2節以降に示す。

3.2 Light WeightなUCD手法

3.2.1 ScrumとUCD手法の対応

UCD手法をScrumに適用する上で、一般的なUCD手法や、UCD手法テーブル[5]の中から、特に代表的な手法で、かつScrumのプロセスに適用できそうな候補を抽出し、Scrumの各フェーズにマッピングしていった。マッピング結果を図 3-1に示す。拡大図は付録 4「ScrumへのUCD手法適用候補」を参照。

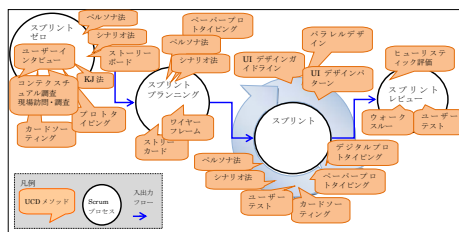


図 3-1 Scrum への UCD 手法適用候補

傾向として、各 UCD 手法の多くは、開発プロセスの全体の前半（スプリントゼロやスプリントプランニング等）にマッピングされた。これは、UCD 手法が上流工程を重視しているためと推察される。

3.2.2 効果的な適用領域の選定

3.2.1節で洗い出したUCD手法の中で、特にScrumの中で効果的に威力を発揮できるUCD手法を選定した。選定結果を図 3-2に示す。拡大図は付録 5「Scrumに対するUCD手法の効果的な適用領域」を参照。

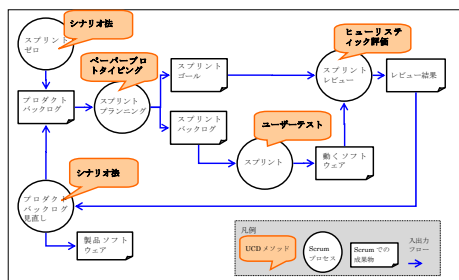


図 3-2 Scrum に対する UCD 手法の効果的な適用領域

各 UCD 手法の選定の理由は以下の通りである。

(1) シナリオ（スプリントゼロ）

シナリオは開発するプロダクトをユーザーがどのように利用するか明確にすることができる。Scrum ではスプリントゼロでプロダクトバックログを作成することになる。その際にシナリオ手法を適用することで、プロダクトの利用シーンを Scrum メンバー内で共有することができ、プロダクトバックログの検討をする上で役立つと考えた。

同様に、Scrum の開発サイクルの中で、プロダクトバックログの見直しをする際も同等の効果を発揮すると考えた。

(2) ペーパープロトタイピング（スプリントプランニング）

ペーパープロトタイピングは開発するプロダクトの完成イメージを早い段階で具体的、かつ、コストを抑えて作成することができる。スプリントに入る前のスプリントプランニングの段階でペーパープロトタイピングを適用することにより、スプリントゴールのイメージをより具体的にすることができる。Scrum メンバー内で開発内容に対する共有認識を持つことで、より開発効率の向上につながると考えた。

(3) ユーザーテスト（スプリント）

ユーザーテストは、効果、効率、満足度に関するユーザビリティの具体的な問題点を、ユーザーが参加する評価を通して把握し、その原因を明らかにすることができる強力なツールである。Scrum 開発サイクルのテスト中に含めることで、利用品質をより高めることができると考えた。ユーザーテストは、他の UCD 手法と比較するとコストがかかるため、スプリントプランニングの段階で実施可否や、ユーザーリクルート等を検討する必要がある。

(4) ヒューリスティック評価（スプリントレビュー）

ヒューリスティック評価は、ユーザビリティに関してユーザビリティ専門家の経験則に基づいた静的で体系的なチェックをすることができる。

ヒューリスティック評価自体は、開発サイクルの中で使用することもできるが、スプリントレビュー時に適用することで、スプリント成果物のユーザビリティに関するレビューが必ず行われることになると考えた。

3.2.3 Light WeightなUCDプロセスの定義

ScrumにUCD手法を適用するにあたり、UCD手法をLight Weight に改善することが重要ではないかと考えた。3.2.2節で選定したUCD手法をScrumに適合するように、軽量化したプロセスを表 3-1の観点で定義した。軽量化した各UCD手法のプロセス内容は、付録 6「Light Weight UCD 手法」を参照。

表 3-1 UCD 手法のプロセス定義表について

プロセスの目的	プロセスを実施する目的や期待する効果に関して記述する。
INPUT	入力成果物に関して記述する。
手順	プロセスの実施手順に関して記述する。
OUTPUT	出力成果物に関して記述する。
条件	プロセスを進める際に必要となる条件に関して記述する。
参加者	プロセスを主体的に進める参加者について記述する。

3.3 適用イメージ

3.2節で定義したプロセスを、3.1節（付録 3）のソフトウェア開発に適用したサンプルイメージを

作成した。詳細は、付録7「工数登録システムへの適用イメージ」を参照。

4 まとめ

4.1 考察

一般に UCD 手法を取り入れることはコストが高くなると思われる。実際、UCD 手法の中にはエンドユーザーを交えて行う必要があるものもある。その結果、最終的には「使いやすいシステム」を構築することができるのだが、コスト削減が求められる昨今、純粋な UCD 手法を開発に組み込むことは難しくなっている。

そのような状況においても、コスト負担がかからないように UCD 手法をカスタマイズすれば、開発に取り入れることができるのではないかと考えた。これが今回の研究の背景にある問題意識であった。

また、近年、「短期間でよいシステムを構築できる」開発手法として、アジャイル開発という開発手法が登場してきた。そこで、今年度は「UCD 手法」と「アジャイル開発」を研究題材として取り上げることとした。

研究を始めた当初、分科会内で「アジャイル的に（＝何度も繰り返す形で）UCD手法を実施する方法」と、「アジャイル開発手法にUCDを取り入れる方法」のどちらを研究するのかという議論があった。3.1節では前者の「アジャイル的にUCD手法を実施する」ことを検討し、実際にある題材について繰り返しUCD手法を実施することを試みた。しかし、「アジャイル的なUCD手法の実施」を行っても、「UCD手法を実施するにあたってのコストを軽減する」という課題を解決することは難しいことがわかった。そこで今回の研究では「アジャイル開発手法にUCDを取り入れる」ことを研究課題とすることとした。

軽量な開発手法として知られているアジャイル開発手法を前提としてそこに UCD 手法を組み込むとすると必然的に UCD 手法自体を軽量化する必要がある。今回の研究ではアジャイル開発手法のアクティビティをベースとし、それに合わせる形で UCD 手法を取り込むこととした。なお取り上げるアジャイル開発手法には Scrum を選択した。

研究の結果、表 4-1のようにUCD手法を軽量化して取り込むためのカスタマイズのポイントを抽出できた。

表 4-1 UCD 手法のカスタマイズ

UCD 手法	Scrum のアクティビティ	カスタマイズのポイント
シナリオ	スプリントゼロおよび各スプリント後に行う、プロダクトバックログの作成・見直し	バックログ抽出を行う際にポイントがわかりやすいようにシナリオの中のキーワードに下線を引く。
ペーパープロトタイプ	スプリントプランニング	「スプリントゴールに対する認識の共有化」というポイントにフォーカスする。
ユーザーテスト	各スプリント	スプリントプランニング時に実施要否を検討し、必要な場合のみ実施する。
ヒューリスティック評価	スプリントレビュー	評価のポイントをチェックリストとして整備し、負荷の軽減、再利用性を図る。

スタンダードな Scrum 開発でも「使いやすいシステム」を構築することは可能であるが、上記のように開発のポイントとなる要所要所に UCD 手法を取り入れることで、効率的に関係者間での合意・共通認識を形成することができると考えられる。通常の Scrum では「動くプログラム」を作り、関係者間での共通認識を形成していくことを考えると、UCD 手法を取り入れたほうがより効率的に「ユーザーが望むもの」に近づけることができると考えられる。

4.2 今後の課題

今回の研究では時間の関係上、実際の開発に適用して検証することができなかった。今後の課題としては、実際に実行することで想定通りの結果を得られるかどうかの検証を行い、今回検討したプロセスについても内容を洗練させ、さらなる軽量化が必要と考える。

また、今回の研究でとりあげた UCD 手法以外の手法の適用や、通常の UCD 手法と軽量化した手法での効果の違いなど、さらに実装まで踏み込んだ研究をしていくこととしたい。

参考文献

- [1] 社)日本事務機械工業会 技術委員会ヒューマンセンタードデザイン小委員会 : "人間中心設計 (ISO13407 対応)プロセスハンドブック <http://www.jbmia.or.jp/~tc/gl-hcd.pdf>", 2.2 節, 社)日本事務機械工業会,2001
- [2] 社)ビジネス機械・情報システム産業協会 技術委員会ヒューマンセンタードデザイン小委員会 : "商品企画フェーズにおける人間中心設計(HCD)プロセスと HCD プロセスの導入効果", 社)ビジネス機械・情報システム産業協会 ,2004
- [3] 樽本徹也 : "ユーザビリティエンジニアリング", オーム社, 2005
- [4] James Kalbach : "デザインング・ウェブナビゲーション", オライリージャパン, 2009
- [5] UsabilityNet : "Usability methods <http://www.usabilitynet.org/tools/methods.htm>"
- [6] Jonathan Arnowitz, Michael Arent, Nevin Berger : "Effective Prototyping for Software Makers", Morgan Kaufmann, 2006
- [7] Desiree Sy : "Adapting Usability Investigations for Agile User-centered Design http://agileproductdesign.com/useful_papers/sy_agile_ucd.pdf", Journal of Usability Studies Vol.2, Issue3, pp.112-132, 2007
- [8] Mike Cohn : "User Stories Applied For Agile Software Development", Addison-Wesley Pearson Education, 2004
- [9] Ken Shewaber, Mike Beedle : "アジャイルソフトウェア開発スクラム", ピアソン・エデュケーション, 2003
- [10] Esther Derby, Diana Larsen : "アジャイルレトロスペクティブズ", オーム社, 2007
- [11] Jim Highsmith : "アジャイルプロジェクトマネジメント", 日経 BP 出版センター, 2005
- [12] JoAnn T. Hackos : "アジャイルな環境における高品質な製品の開発 : http://www.designit.jp/archives/cat74/agile_ucd/agile_software_development_j.pdf", Comtech Services, Inc, 2009
- [13] Mike Cohn : "アジャイルな見積りと計画づくり",毎日コミュニケーションズ, 2009
- [14] Venkat Subramaniam, Andy Hunt : "アジャイルプラクティス", オーム社, 2007
- [15] Kent Beck : "XP エクストリームプログラミング入門", ピアソン・エデュケーション 2005
- [16] James Shore, Shane Warden : "アート・オブ・アジャイルデベロップメント", オライリージャパン, 2009
- [17] Ken Schewaber, Jeff Stutherland : "SCRUM GUIDE <http://www.scrum.org/storage/scrumguides/Scrum%20Guide.pdf>", Scrum.org, 2008
- [18] Scrum Alliance : <http://www.scrumalliance.org/>

付録1 アジャイル

アジャイル開発の定義

アジャイルとは「すばやい=agile」という意味の英単語であり、開発手法の意味としては、現状の変化の激しいビジネス環境に「すばやく」対応できる軽量で柔軟、且つ効率的なシステム開発手法を指す。

Manifesto for Agile Software Development (<http://agilemanifesto.org/> 日本語訳：清水)

アジャイルソフトウェア開発マニフェスト

我々はソフトウェア開発のより良い方法を探し求めている。
探し求める過程で、以下の価値を持つに至った：

プロセスやツールよりも個人の能力とメンバー間での交流を
ドキュメントを完成させることよりも動くソフトウェアを
契約交渉よりも顧客との相互協力を
計画に従うことよりも変化への対応を

上記の項目では、我々は前者の項目よりも後者の項目により
重きを置いている。

Scrum の用語

(「実践！アジャイルプロジェクト管理」、及び「SCRUMGUIDE(<http://www.scrum.org>)」より)

1. プロダクトバックログ
プロダクトが満たすべきであろう全ての要求項目が、優先度付けされて記載されたリスト。
様々な粒度の要求項目を記載可能。
2. スプリントゼロ
プロダクトバックログを作成する初期フェーズ。
3. スプリントゴール
1回のスプリントで満たすべきプロダクトバックログの要求項目。
4. スプリントバックログ
プロダクトバックログ（スプリントゴール）を実装する上で、動くソフトウェアとして1回のスプリントで実装可能なタスクリスト。
5. スプリントプランニング
Scrum で定義されたタイムボックスの1つで、スプリント計画会議とも言う。スプリントの開始時に毎回開催し、スプリントゴールとスプリントバックログを作成する。通常1ヶ月のスプリントを行う時には、8時間のスプリントプランニングを行う（おおよそ全体スプリント期間の5%とされている）。8時間の内、前半の4時間で何を実装するかを決定し、後半の4時間で各々の開発メンバーがどの様に実装するかを話しあう。
6. スプリントレビュー
Scrum で定義されたタイムボックスの1つで、スプリントの終了時スクラムチームとステークホルダーが協力して開催する。スプリント自体のレビューとスプリント期間に発生した

プロダクトバックログへの変更について話し合い、次に何ができるかを検討する。通常1ヶ月のスプリントを行うときには、4時間のスプリントレビューを行う（全体スプリント期間の5%以上の時間を使ってはいけない）。スプリントレビューでは、およそ以下の4つの内容を話しあう。

- A) プロダクトオーナーは何がすでになされ、何がまだなされていないかを識別する。
- B) スプリントチームは何がうまく行って何が問題として残っているか、そしてどのように解決するかについて話しあう。
- C) スプリントチームは様々な条件で製品を動かして質問に答え、プロダクトオーナーはそのプロダクトバックログのあらわす内容について話しあう。
- D) メンバー全体で何がなされ、次に何がなされるべきなのかを話しあう。

付録2 UCD

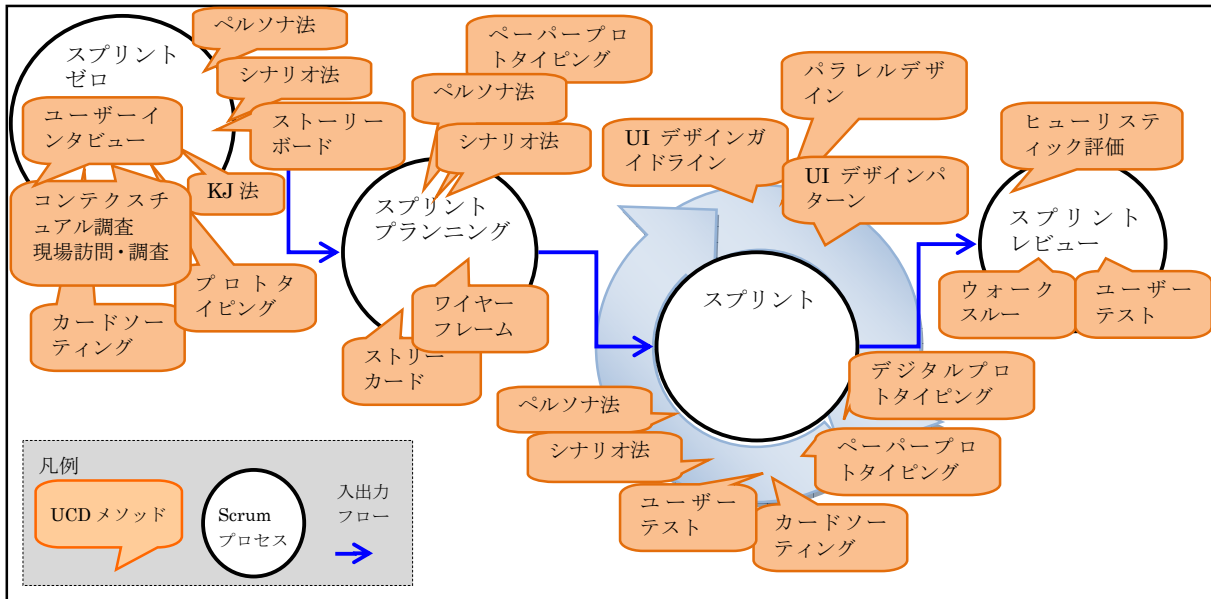
ISO 9241-11(JIS Z 8522)によるユーザビリティ定義

ある製品が、指定された利用者によって、指定された利用の状況下で、指定された目的を達成する際の、有効さ、効率及び利用者の満足度の度合い。

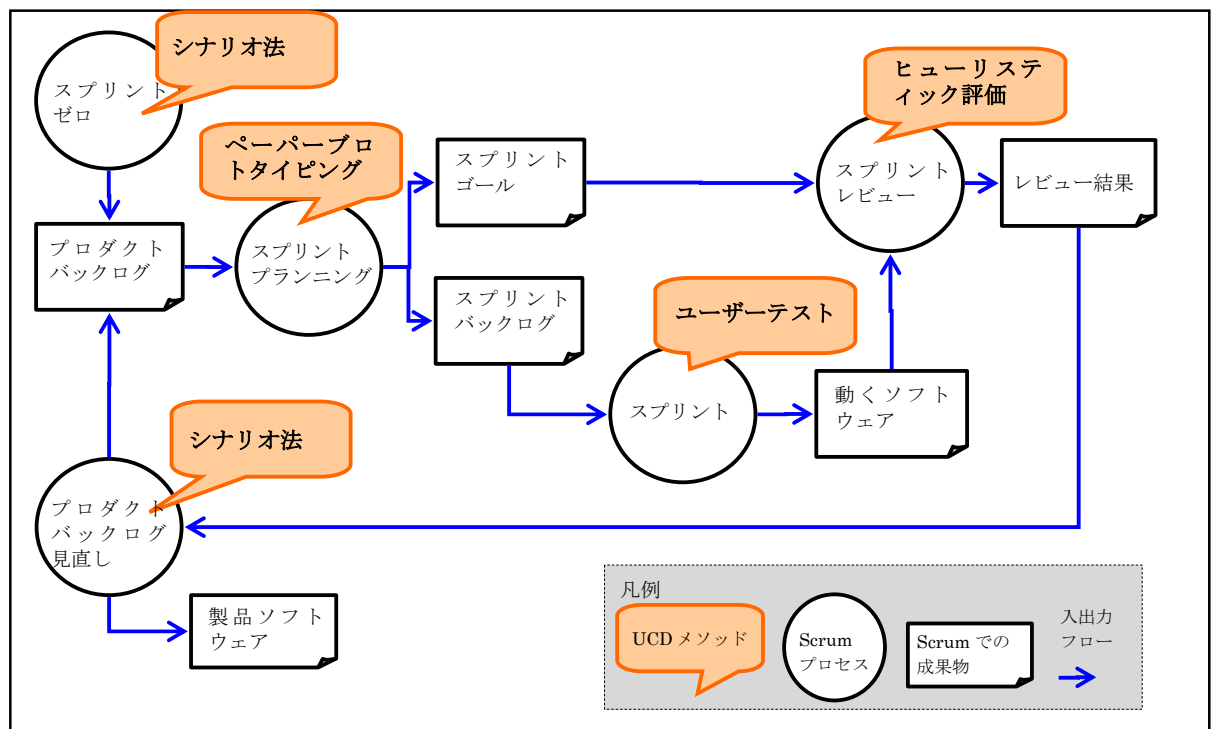
付録3 工数登録システム

CSV出力				
工数登録		2009年6月		
	社員番号	1T5331		
	名前	林田 一郎		
日付	案件名(プロジェクト名)	作業内容	作業時間(H)	備考
7/10	AAA案件/2009010-F1	問合せ対応	0.50	
7/10	AAA案件/2009010-F1	インシデント管理	0.25	
7/10	AAA案件/2009010-F1	問合せ調査	3.60	
7/10	AAA案件/2009010-F1	調査報告	0.25	
7/10	BBB案件/2009050-F2	内部ミーティング	1.00	
7/10	BBB案件/2009050-F2	保守改修	2.00	
7/10	事務処理/その他	メール確認	0.50	
7/10	事務処理/通常業務	工数登録へアップロード	0.30	
7/10	事務処理/日報	日報報告	0.10	

付録4 Scrumに対するUCD手法適用候補



付録5 Scrumに対するUCD手法の効果的な適用領域



付録6 Light Weight UCD手法

付録6.1 シナリオ

プロセス名称	シナリオ
プロセスの目的	物語風の文章をもとにユーザーの行動やシステムの問題点を洗い出し、活用する。
INPUT	・ペルソナ もしくは ユーザー調査結果
手順	<ol style="list-style-type: none"> 1.シナリオの作成 <ol style="list-style-type: none"> 1.1 業務を時系列、物語風に記述する 1.2 記述した物語を日常、必須、エッジに分類する 2.シナリオの分析 <ol style="list-style-type: none"> 2.1 プロダクトバックログの要素となりうるキーワードに下線を付ける 2.2 プロダクトバックログの重要度と分類（日常、必須、エッジ）を対応づける
OUTPUT	・プロダクトバックログの要素が明確になったシナリオ
条件	<ul style="list-style-type: none"> ・業務を理解している人がシナリオ作成に参加していること ・プロダクトオーナーが参加していること
参加者	プロダクトオーナー、スクラムチームメンバー

付録6.2 ペーパープロトタイピング

プロセス名称	ペーパープロトタイピング
プロセスの目的	スプリントに入る前に、ゴールのイメージを共有するスプリントプランニングを助ける。
INPUT	<ul style="list-style-type: none"> ・プロダクト バックログ (全体像に立ち返る場合はシナリオやペルソナも適宜使用する)
手順	<ol style="list-style-type: none"> 1.プロダクト バックログからスプリントで実施するものを選択する 2.ペーパープロトタイピングの素材を準備する 3.ペーパープロトタイピングを作成し、動かす <ol style="list-style-type: none"> 3.1 スプリントのタスク、課題を抽出する 3.2 スプリント ゴールを精緻化する 4.スプリントプランニングを行う
OUTPUT	<ul style="list-style-type: none"> ・合意されたペーパープロトタイピングの振る舞い ・実装するバックログの動きに対しての共通認識
条件	
参加者	プロダクトオーナー、スクラムチームメンバー

付録6.3 ユーザーテスト

プロセス名称	ユーザーテスト
プロセスの目的	効果、効率、満足度に関するユーザビリティの具体的な問題点をユーザーが参加する評価を通して把握し、その原因を明らかにする。 ユーザーに評価対象を操作して予め設定したタスクを実行してもらっている過程の発話からユーザーインターフェースの問題点を明確にする。
INPUT	<ul style="list-style-type: none"> ・ペルソナ もしくは ユーザー調査結果 ・シナリオ
手順	<p>1.テスト設計</p> <p>1.1 テスト対象の主要タスクを選別する</p> <p>1.2 テストシナリオ作成 主要タスクのテストシナリオを作成する ※シナリオがそのまま流用可能であれば作成は不要</p> <p>2.テスト実施</p> <p>2.1 テストスクリプトの作成</p> <p>2.2 インタビューリストの作成</p> <p>2.3 テスト実施（思考発話法）</p> <p>2.4 満足度に関するインタビュー</p>
OUTPUT	<ul style="list-style-type: none"> ・ユーザーテスト分析レポート ユーザビリティの問題点とその詳細内容、重要度、改善点等のリスト ・ユーザーテストログ
条件	<ul style="list-style-type: none"> ・テスト実施可能な動作するソフトウェア ・スプリントプランニングでテスト計画を実施する。 <p>スプリントプランニング時にはプロダクトオーナーとともにユーザーテストの実施を判断する。</p> <p>①このスプリントでユーザーテストを行うかどうか ペーパープロトタイプング通してユーザーテストが必要かどうかを決める</p> <p>②ユーザーテストを行う場合は手順をスプリントバックログに入れる 手順1について明確な場合は実施不要。手順2は必須。</p> <p>③ユーザーのリクルート</p> <p>④テスト実施場所（ユーザビリティラボ等）の確保</p>
参加者	ユーザー、スクラムチームメンバー

付録6.4 ヒューリスティック評価

プロセス名称	ヒューリスティック評価
プロセスの目的	スプリントレビュー時に、ユーザビリティ評価法を利用してスプリント成果物がスプリントゴール（ユーザビリティに関して）に適合するかのレビューを行う。
INPUT	<ul style="list-style-type: none">・スプリントゴール・スプリント成果物（動くソフトウェア）
手順	1.ヒューリスティック評価 1.1 スプリント成果物を評価基準に照らして、レビューを実施する 1.2 結果を報告書にまとめる →上記でプロダクトバックログを変更すべき点は候補としてまとめた報告書を作成する。
OUTPUT	<ul style="list-style-type: none">・レビュー結果（ヒューリスティック評価報告書）
条件	<ul style="list-style-type: none">・業務を理解している人が参加していること・プロダクトオーナーが参加していること・作業規模・期間、及び予算によって適応深度を設定
参加者	プロダクトオーナー、スクラムチームメンバー

付録7 工数登録システムへの適用イメージ

