日本デザイン学会 デザイン学研究

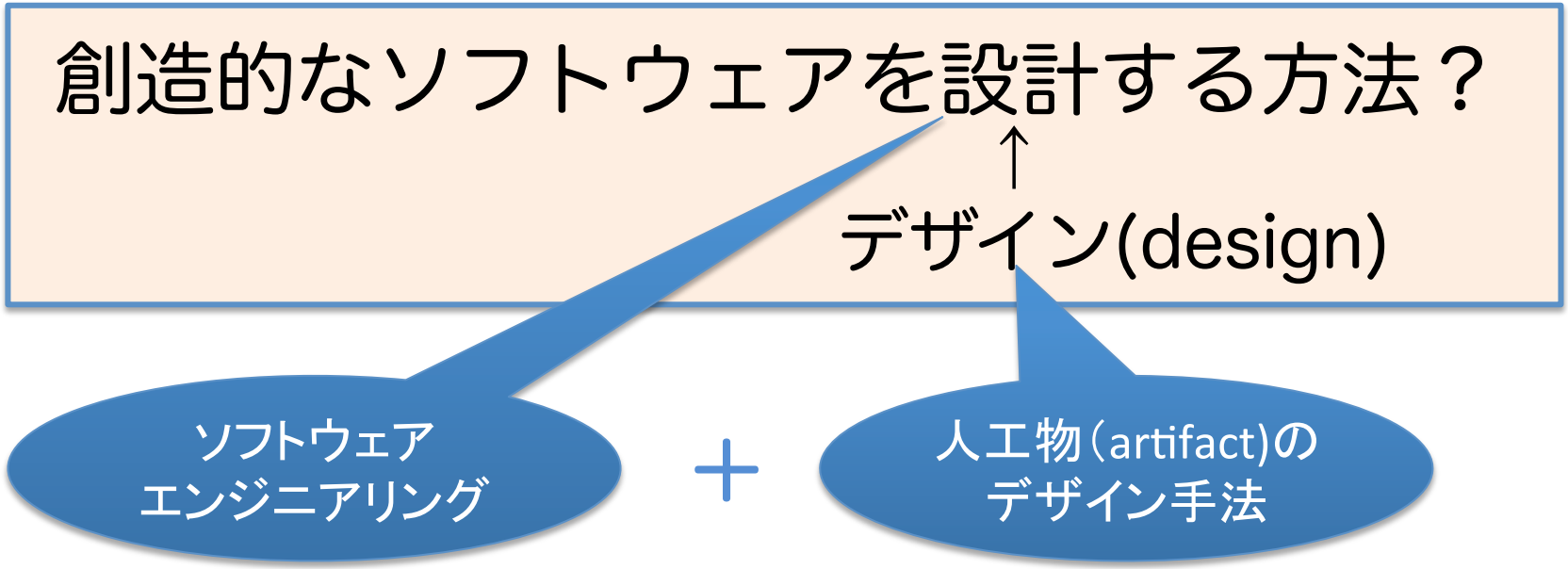
実行可能知識としてのソフトウェア構築プロセス

2012年6月23日

大槻 繁　株式会社一（いち）

前世紀まで、重厚長大伝統的大企業で
ソフトウェアエンジニアリングの研究・開発

今世紀は、軽量俊敏な小さいコンサルティングファーム
を経営しながら、新しいソフトウェア作りを探求



起 ソフトウェアエンジニアリングの呪縛

承 新ソフトウェア宣言

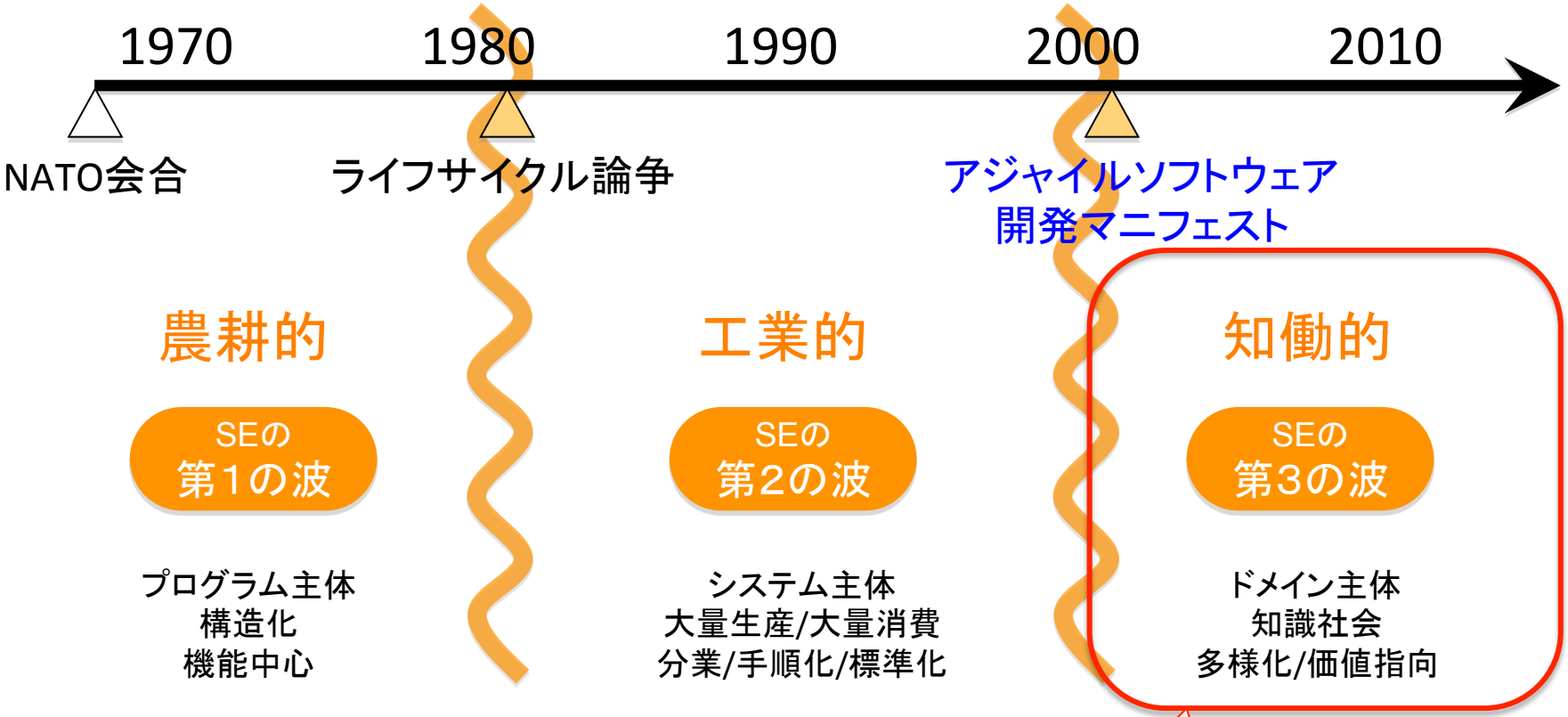
転 ΛVモデル

結 アーティファクトとしてのソフトウェア

起

ソフトウェアエンジニアリングの呪縛

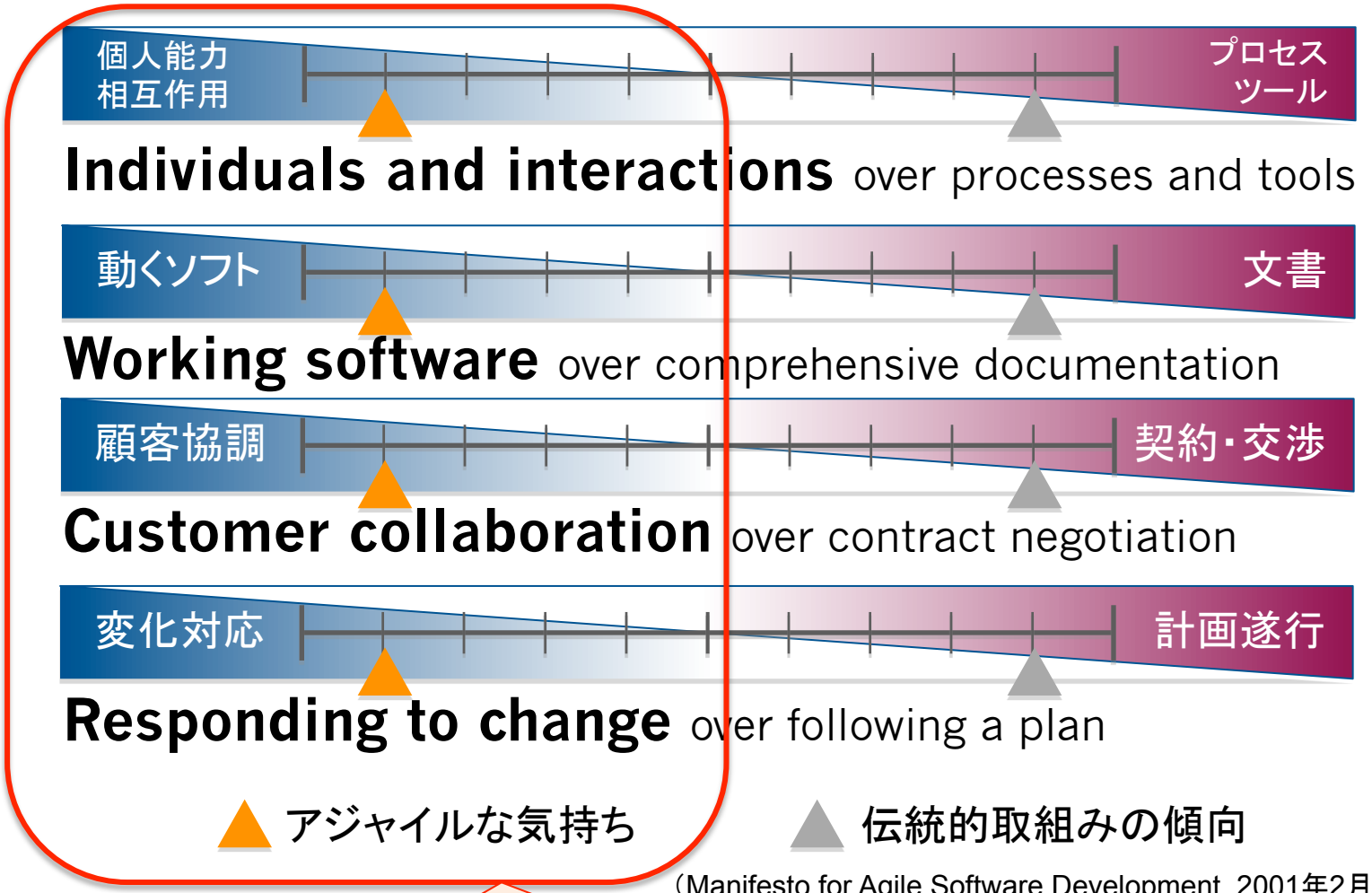
ソフトウェアエンジニアリングのあゆみ：第1～3の波



アルビン・トフラーの『第三の波』で
提示されたパラダイムを、ソフトウェアの
世界にあてはめてみました。

時代は「第3の波」なのに、
「第2の波」に呪縛されている

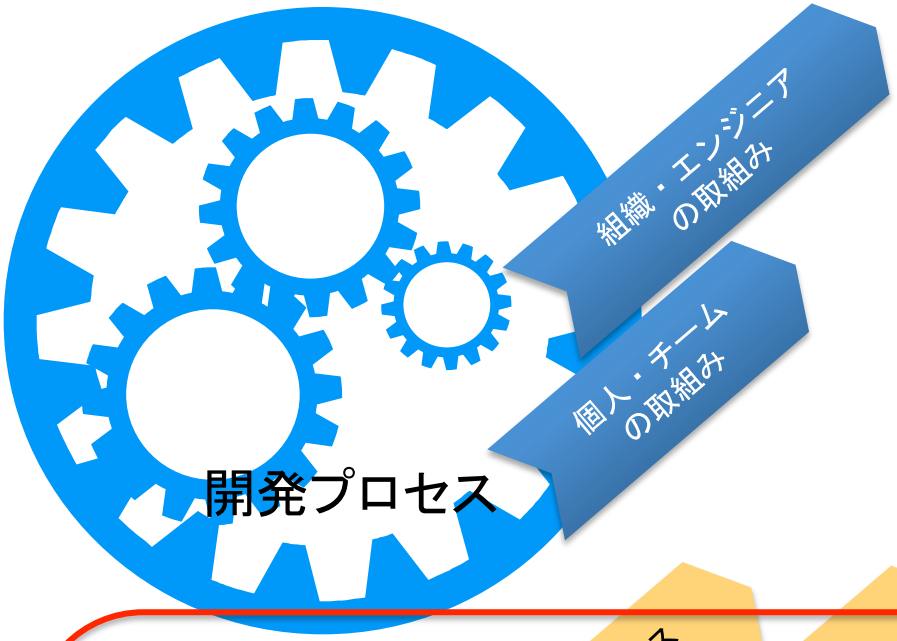
アジャイルソフトウェア開発マニフェストは、それなりにインパクトがありました・・・



（Manifesto for Agile Software Development, 2001年2月）

ソフトウェア作りで大切なこと
を改めて確認

狭義から広義のアジャイルへ



狭義のアジャイルプロセス

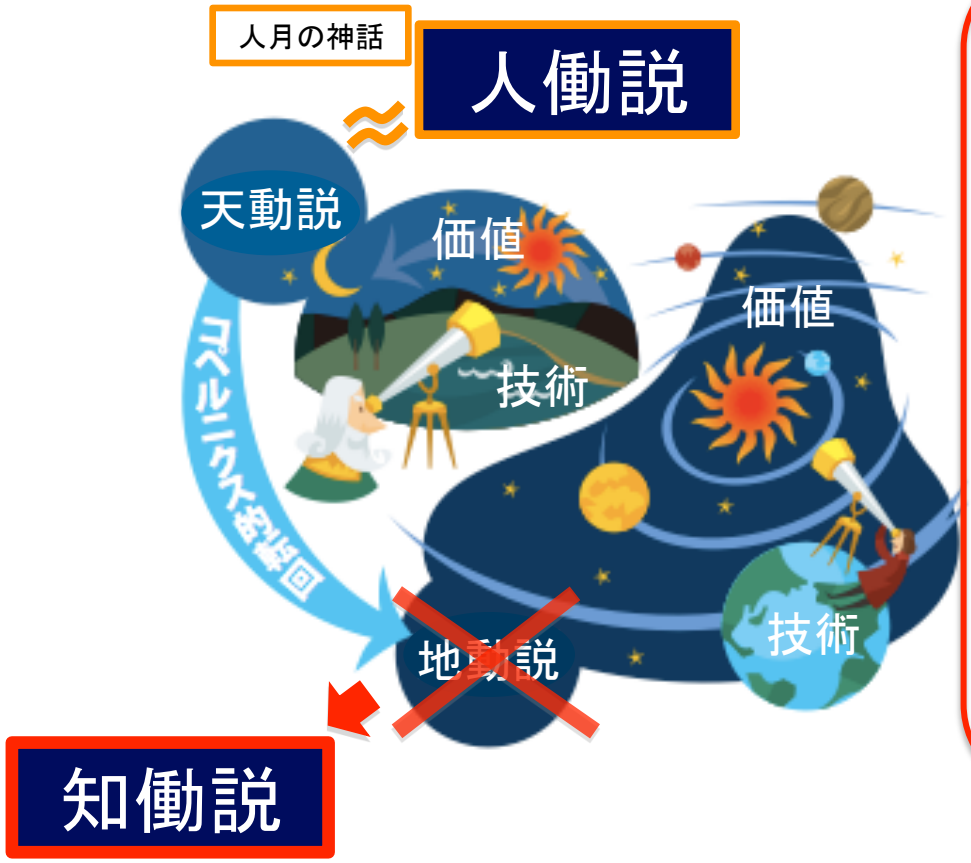
開発者側中心の視点
顧客側とのコミュニケーションや確認を重視してはいるものの、あくまでも受動的

アジャイルプロセスの取り組みは、最近ではビジネス連動が中心課題

広義のアジャイルプロセス

顧客側と開発側との同期
全体での価値創出、ビジネスプロセスを能動的に考慮

人働説から知働説へ



「作る」と「使う」は本質的に同じ

◆ソフトウェアとは
《実行可能な知識》である
Executable Knowledge

◆ソフトウェアとは
実行可能な知識を紡いだ
《様相》である
Texture

◆中心は《機能》から《様相》へ

知働化研究会は、アジャイルプロセス協議会のWGとして、2009年6月に設立されました。
コンセプトリーダ：山田正樹氏（メタボリクス社）
運営リーダ：大槻繁、現在メンバ数24、研究室数6
http://www.exekt-lab.org/

知働化研究会では、時々、研究誌をまとめることにしています。
創刊第1号が2010年11月3日（文化の日）に公開されました。
http://www.exekt-lab.org/Home/exerev

知働化研究会誌Volume 1は、知働化研究会サイトから無料でダウンロードできます。面白い作品が満載です。お楽しみください。
現在、Volume 2の発行に向けて、執筆や準備を進めています。

承

新ソフトウェア宣言

新ソフトウェア宣言

Declaration of New Software

従来のソフトウェアに関わる諸活動の呪縛から解放され、新たな世界を築いていくためには、新たな呪縛に捕われ、新たな中心を設定していかなくてはなりません。我々は、以下の関心の候補が有望であると確信しています。

In order to build a new world, freed from the curse of activities related to traditional software field, we shall be caught in a new curse, it does go to a new paradigm will set. We are strongly confident that the following are promising candidates for the matter of concern.

1．ソフトウェアは、数学的理論探求の上に成り立つ
Software is built upon exploration of mathematical theory.

2．ソフトウェアは、部分に還元することが不可能な全体である
Software can not be reduced to parts.

3．ソフトウェアは、実行可能な知識である
Software is executable knowledge.

4．ソフトウェアは、学びの副産物に過ぎない
Software is just a by-product of learning.

5．ソフトウェアは、制約条件下で創造される美しい人工物である
Software is a beautiful artifact, created under some constraints.

6．ソフトウェアは、富を生む経済活動の資源である
Software is a resource of wealth-creating economic activities.

7．ソフトウェアは、言語ゲームである
Software is a language game.

上記、基盤化、全体論、知働化、進化論、意匠論、経営論、遊戯論などの広範で、かつ、相互補完的アプローチが、新しい物語を生み出し、新たなる地平を切り開いていくことでしょう。

Above, the mathematical infrastructure, Holism, executable knowledge / texture, evolution, design theory, management theory, and language game, those are broad and the complementary approaches, creating a new story, we will open up a new horizon.

大槻繁，芝元俊久，高野明彦，竹内雅則，時本永吉，夏目和幸，萩原正義，羽生田栄一，濱勝巳，本橋正成，山田正樹，綿引琢磨

Shigeru Otsuki, Shibamoto Toshihisa, Akihiko Takano, Masanori Takeuchi, Eikichi Tokimoto, Kazuyuki Natsume, Masayoshi Hagiwara, Eiichi Hanyuda, Katsumi Hama, Masanari Motohashi, Masaki Yamada, Takuma Watabiki

Agile Process Association　知働化研究会

2010 年 6 月 11 日 横浜にて
June 11, 2010 at Yokohama

『新ソフトウェア宣言』は、通称『呪縛宣言』とも呼ばれています。

2010年6月9日～11日に横浜開港記念館で開催されたソフトウェアシンポジウム2010の中のワーキンググループ「ソフトウェアエンジニアリングの呪縛WG」に集まった賢人たちが、これからのソフトウェア、および、ソフトウェアエンジニアリングの方向性について議論したものが元になっています。

最終的には、「呪縛WG」の二人のコーディネータである大槻繁と濱勝巳氏によって、WG終了後議論が重ねられ、今の7項目からなる形に集約されました。

http://www.exekt-lab.org/Home/newsoftdecl

iPad用のデザインも提供
飾っておくとじわじわっと効いてくる

新ソフトウェア宣言

従来のソフトウェアに関わる諸活動の呪縛から解放され、新たな世界を築いていくためには、新たな呪縛に捕われ、新たな中心を設定していかなくてはなりません。我々は、以下の関心の候補が有望であると確信しています。

1. ソフトウェアは、数学的理論探求の上に成り立つ

2. ソフトウェアは、部分に還元することが不可能な全体である

3. ソフトウェアは、実行可能な知識である

4. ソフトウェアは、学びの副産物に過ぎない

5. ソフトウェアは、制約条件下で創造される美しい人工物である

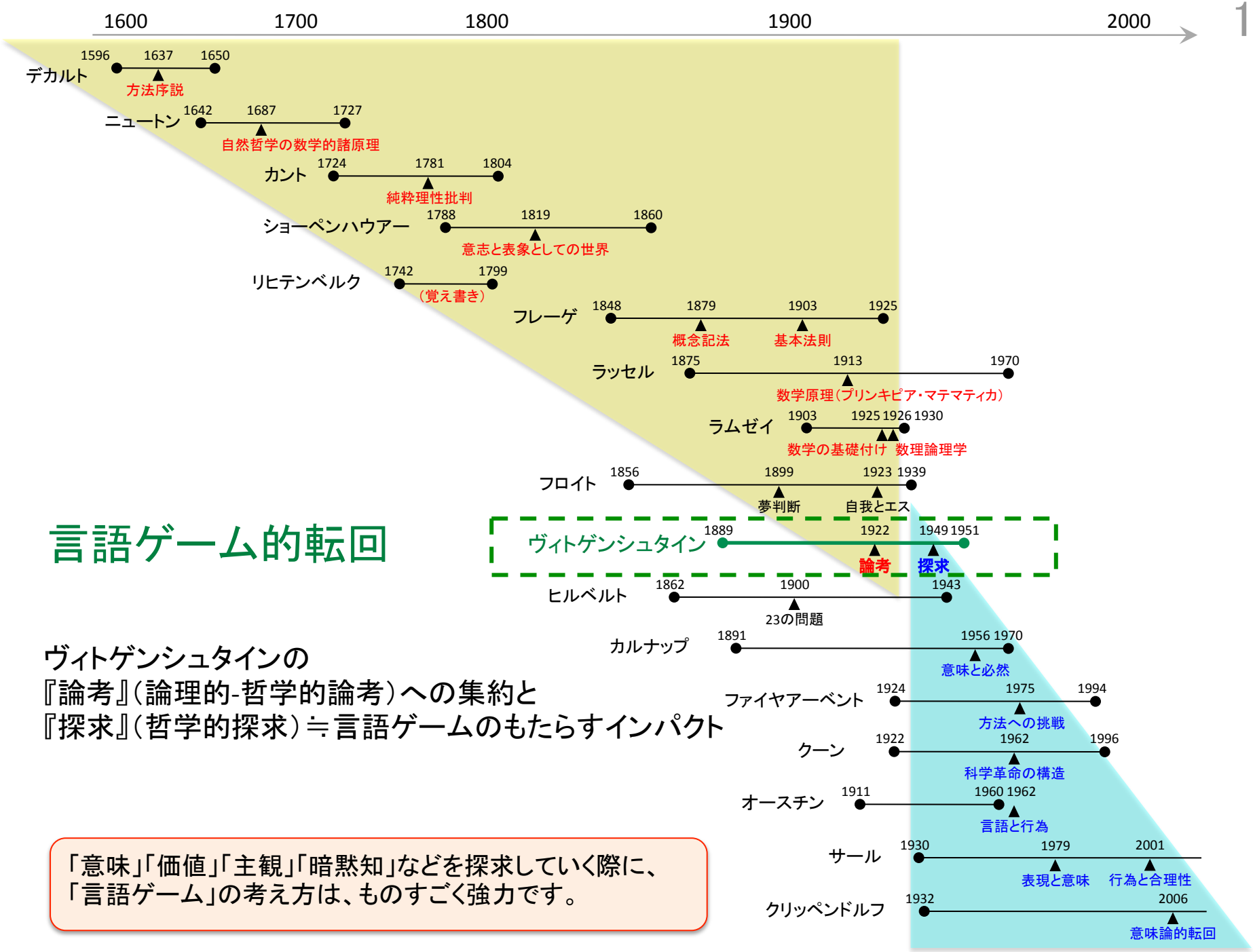
6. ソフトウェアは、富を生む経済活動の資源である

7. ソフトウェアは、言語ゲームである

上記、基盤化、全体論、知働化、進化論、意匠論、経営論、遊戯論などの広範で、かつ、相互補完的アプローチが、新しい物語を生み出し、新たなる地平を切り開いていくことでしょう。

大槻繁，芝元俊久，高野明彦，竹内雅則，時本永吉，夏目和幸，萩原正義，
羽生田栄一，濱勝巳，本橋正成，山田正樹，綿引琢磨

2010年6月11日 横浜にて
June 11, 2010 at Yokohama



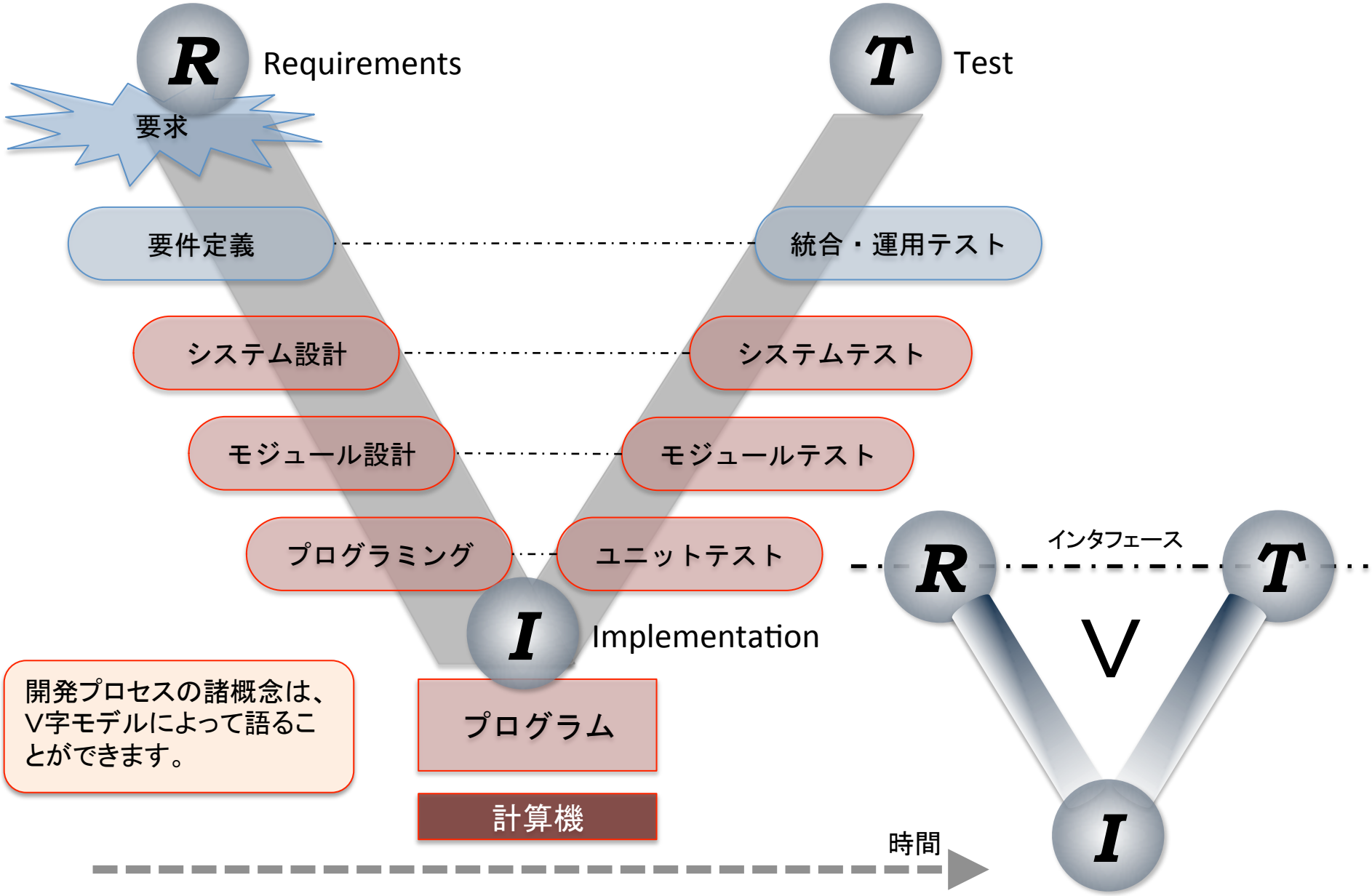
言語ゲーム的転回

ヴィトゲンシュタインの
『論考』（論理的-哲学的論考）への集約と
『探求』（哲学的探求）≒言語ゲームのもたらすインパクト

「意味」「価値」「主観」「暗黙知」などを探求していく際に、
「言語ゲーム」の考え方は、ものすごく強力です。

転

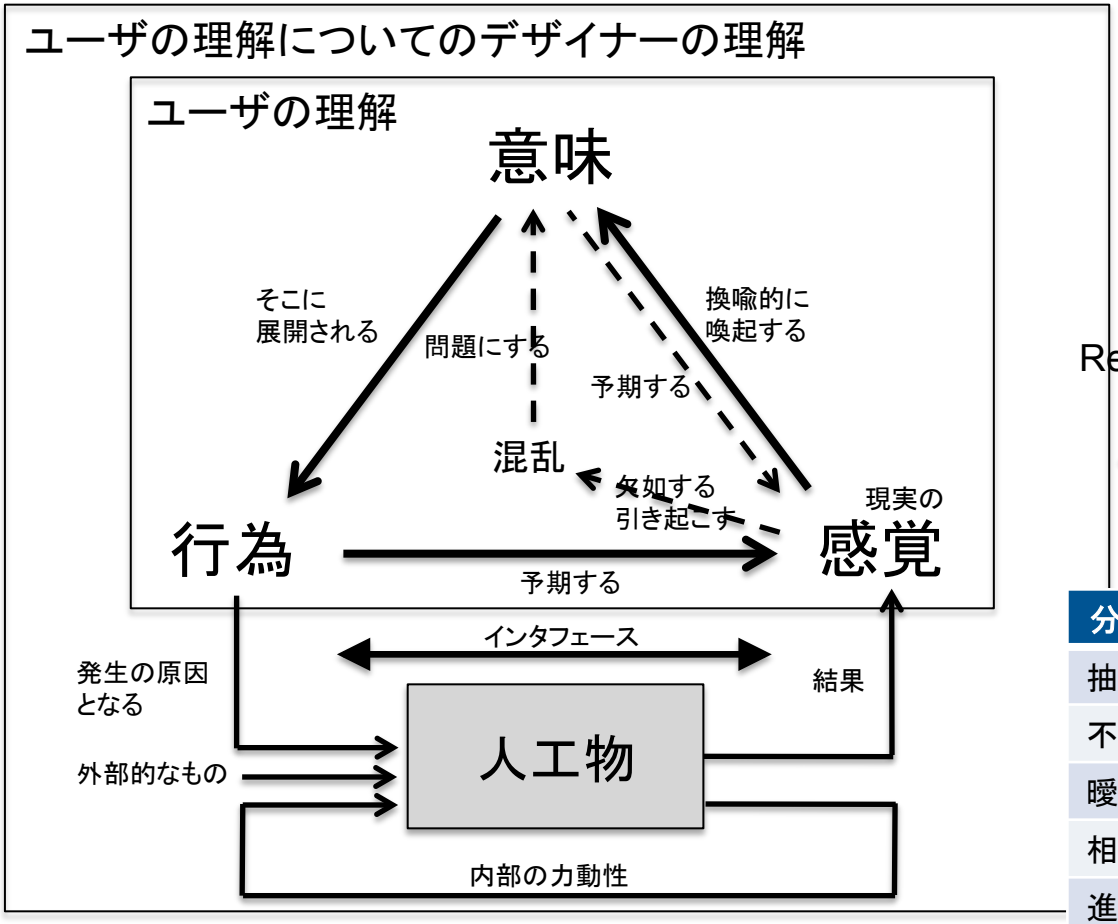
ＡＶモデル

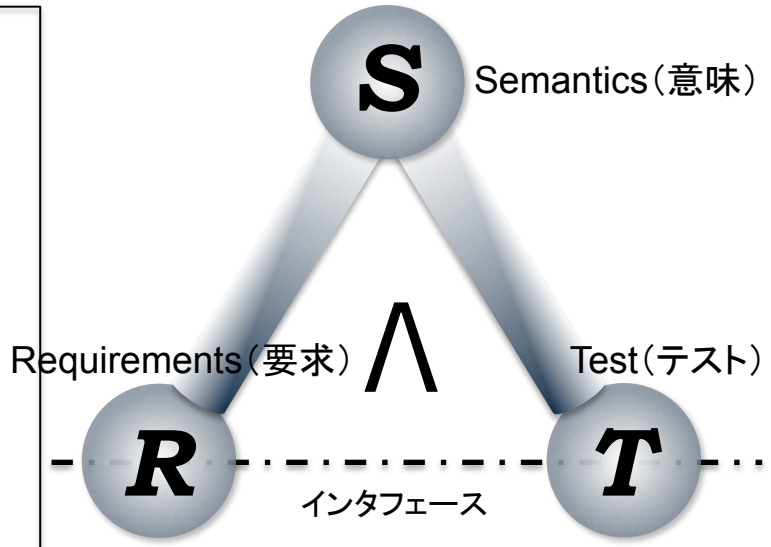

R
Requirements
要求
T
Test
要件定義
統合・運用テスト
システム設計
システムテスト
モジュール設計
モジュールテスト
プログラミング
ユニットテスト
I
Implementation
プログラム
計算機
開発プロセスの諸概念は、V字モデルによって語ることができます。
時間
R
インタフェース
T
V
I

ソフトウェアのデザイン

デザイン論の大御所であるクリッペンドルフは著作『意味論的転回』の中で、左図のような意味、行為、感覚の関係を提唱しています。
これは、ソフトウェアをデザインの対象とした時にも成立します。

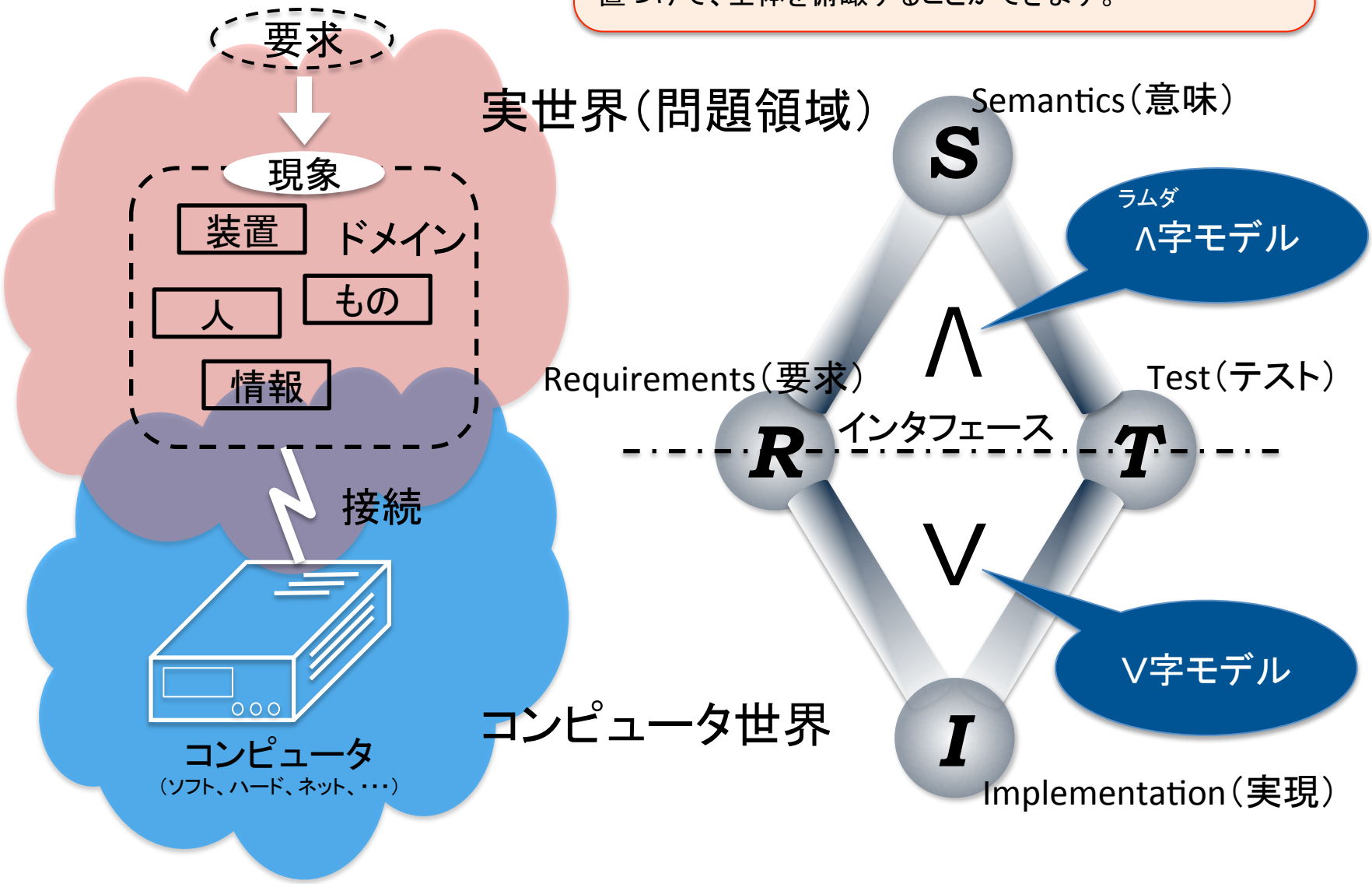
デザイナがユーザのインタフェースについて持つ理解



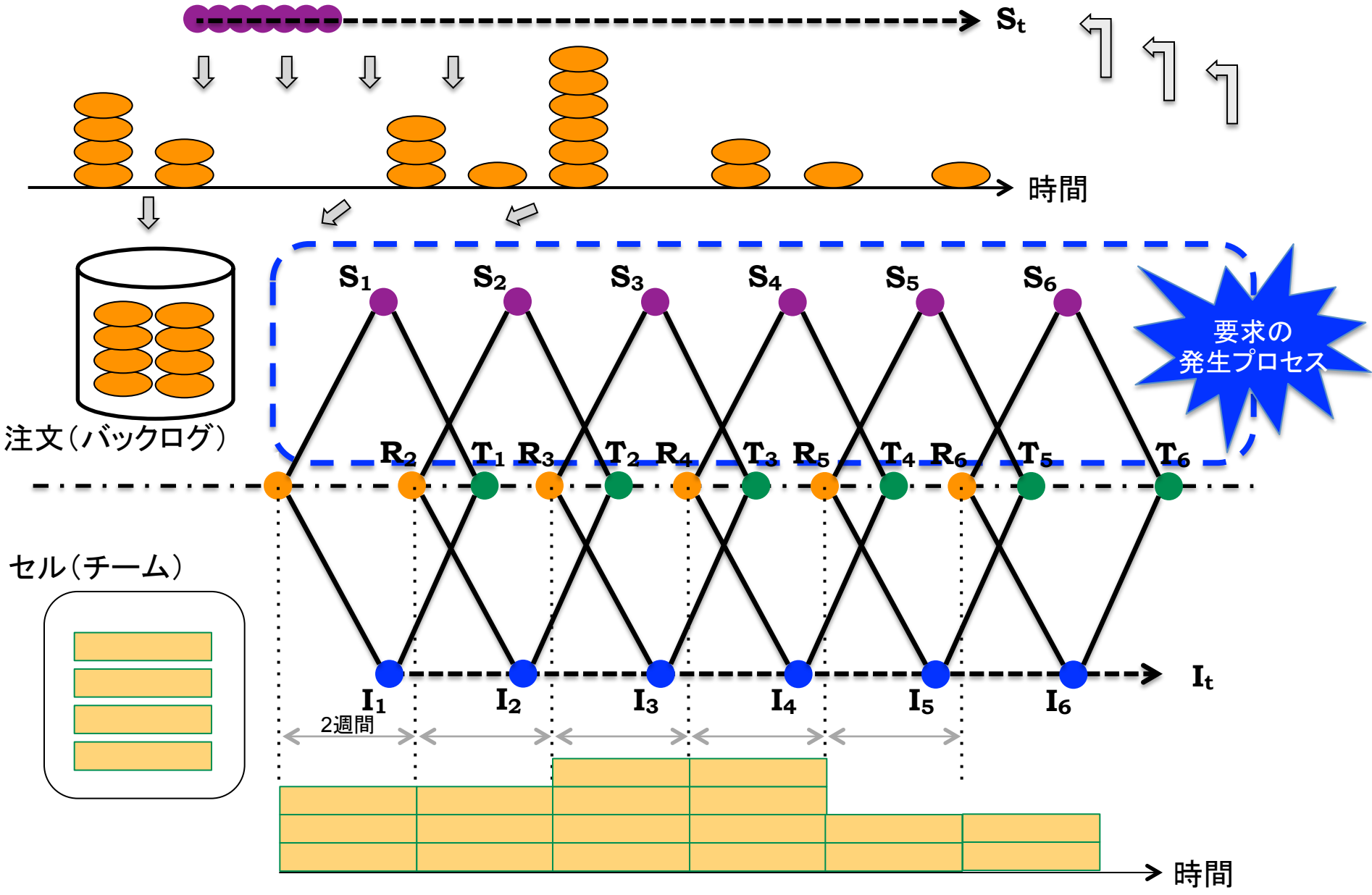


分類名称	説明	典型的対象分野
抽象的	高設計自由度	全般
不確実	確率・非決定的	予測、フィルタ
曖昧	未認識・未定義対応	全般
相対的	言語ゲーム的	社会、対話
進化的	メタ・状況適応	パターン認識
計算限界	近似解	レイアウト、検索
未解明項	調整・フィードバック	制御、ログ解析

V字モデルとデザイン論の枠組みを接合することによって、『ΛVモデル』ができあがります。これは、ソフトウェア（システム）、対象領域（実世界）、要求との関係も明確に位置づけて、全体を俯瞰することができます。



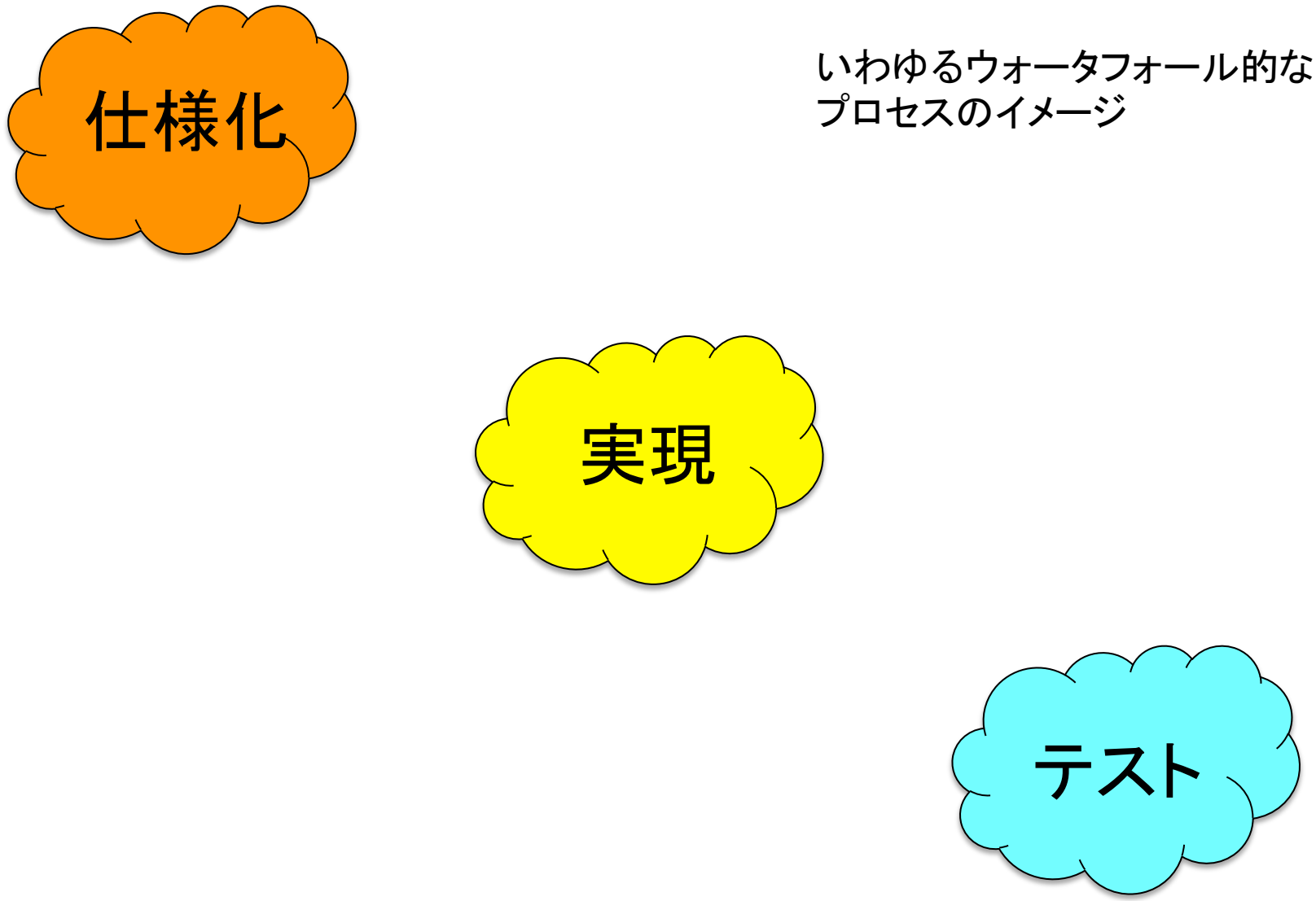
アジャイルプロセスのタイムボックス方式（ソフトウェアセル生産方式）をＡＶモデルで分析することができます。要求の発生プロセス（受注生産であればクライアント側）を明確にしていく必要があります。



結

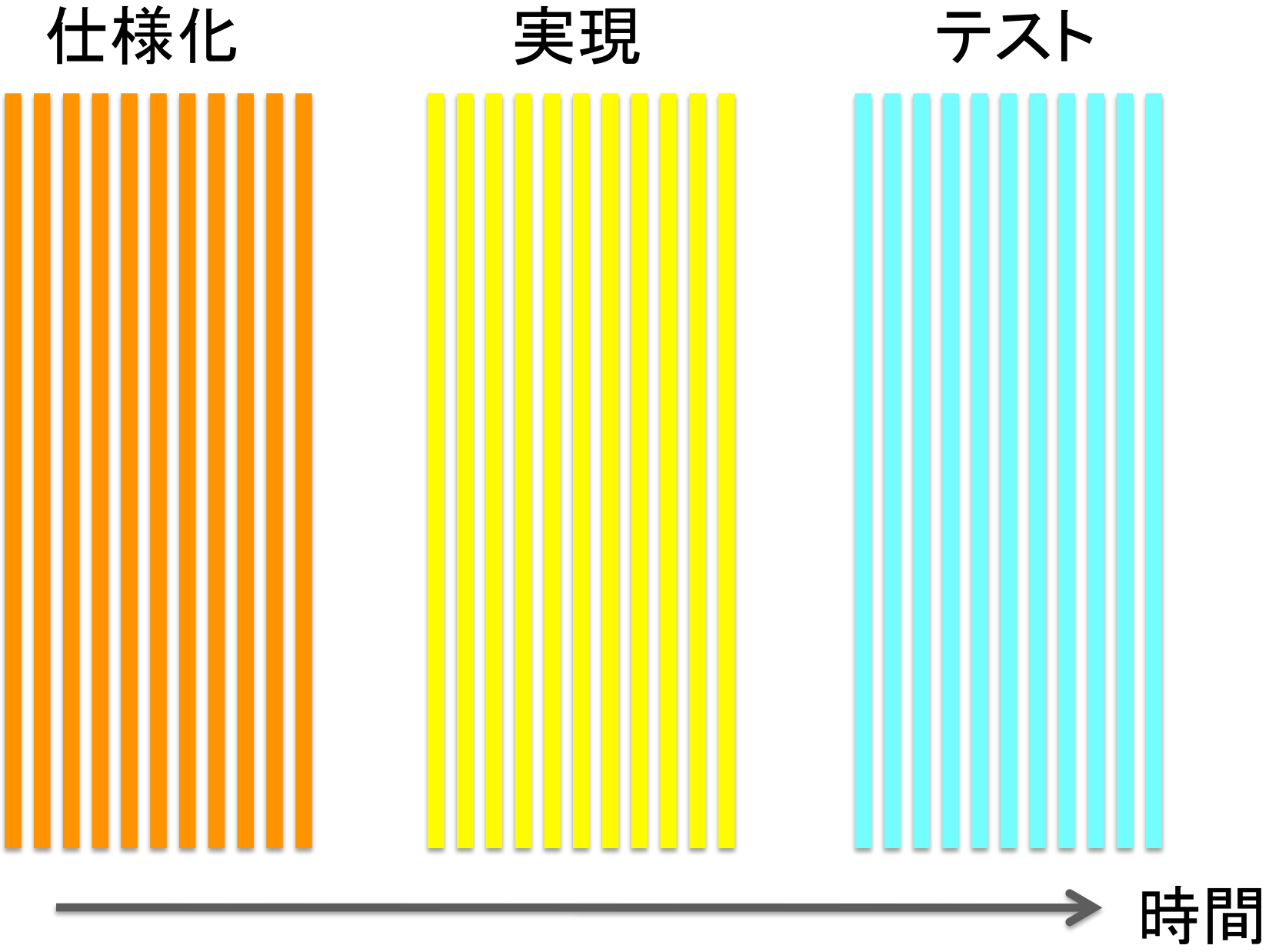
アーティファクトとしてのソフトウェア

いわゆるウォータフォール的な
プロセスのイメージ



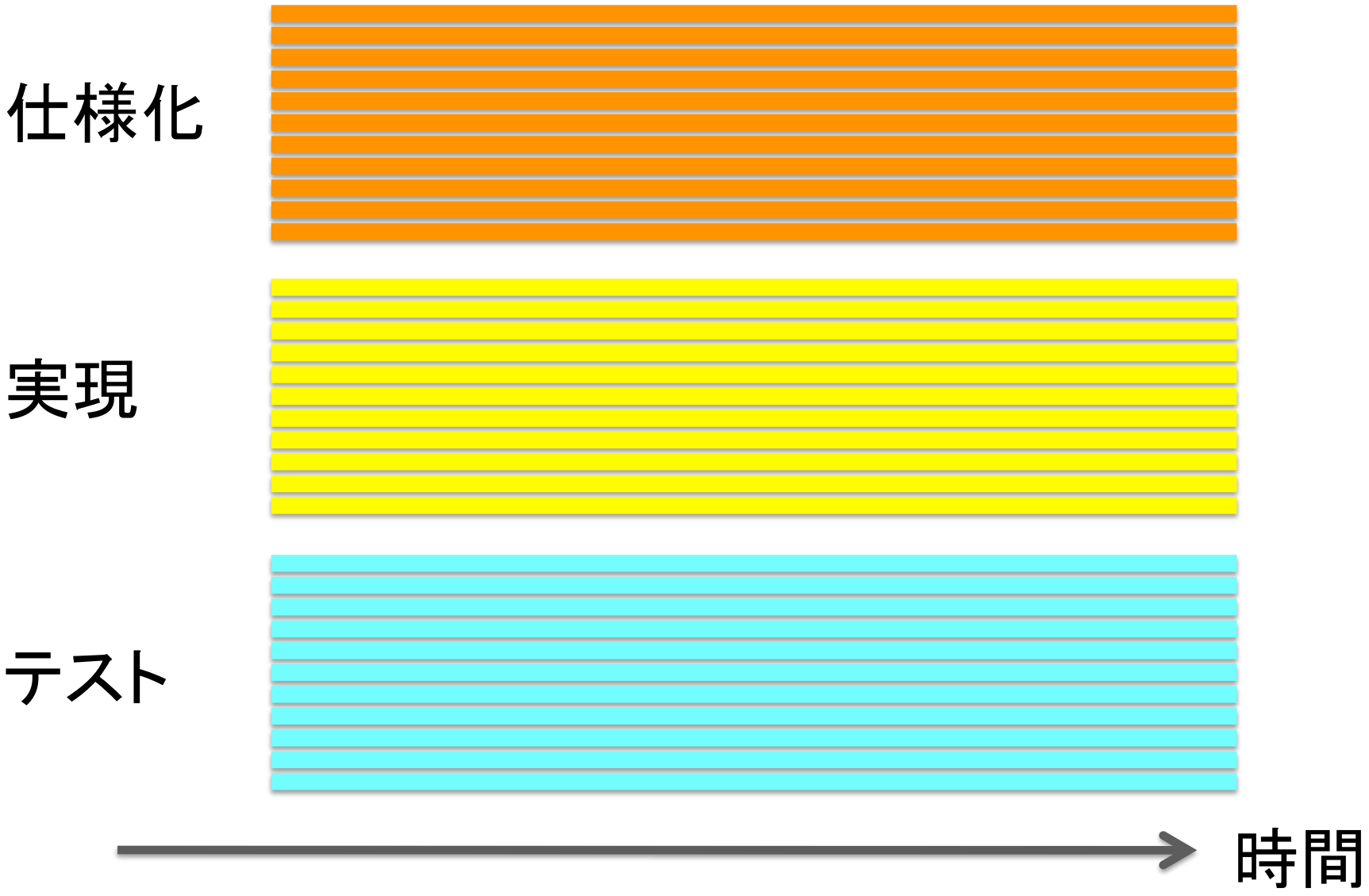
逐次(順次)プロセス

「;」は、逐次オペレータ

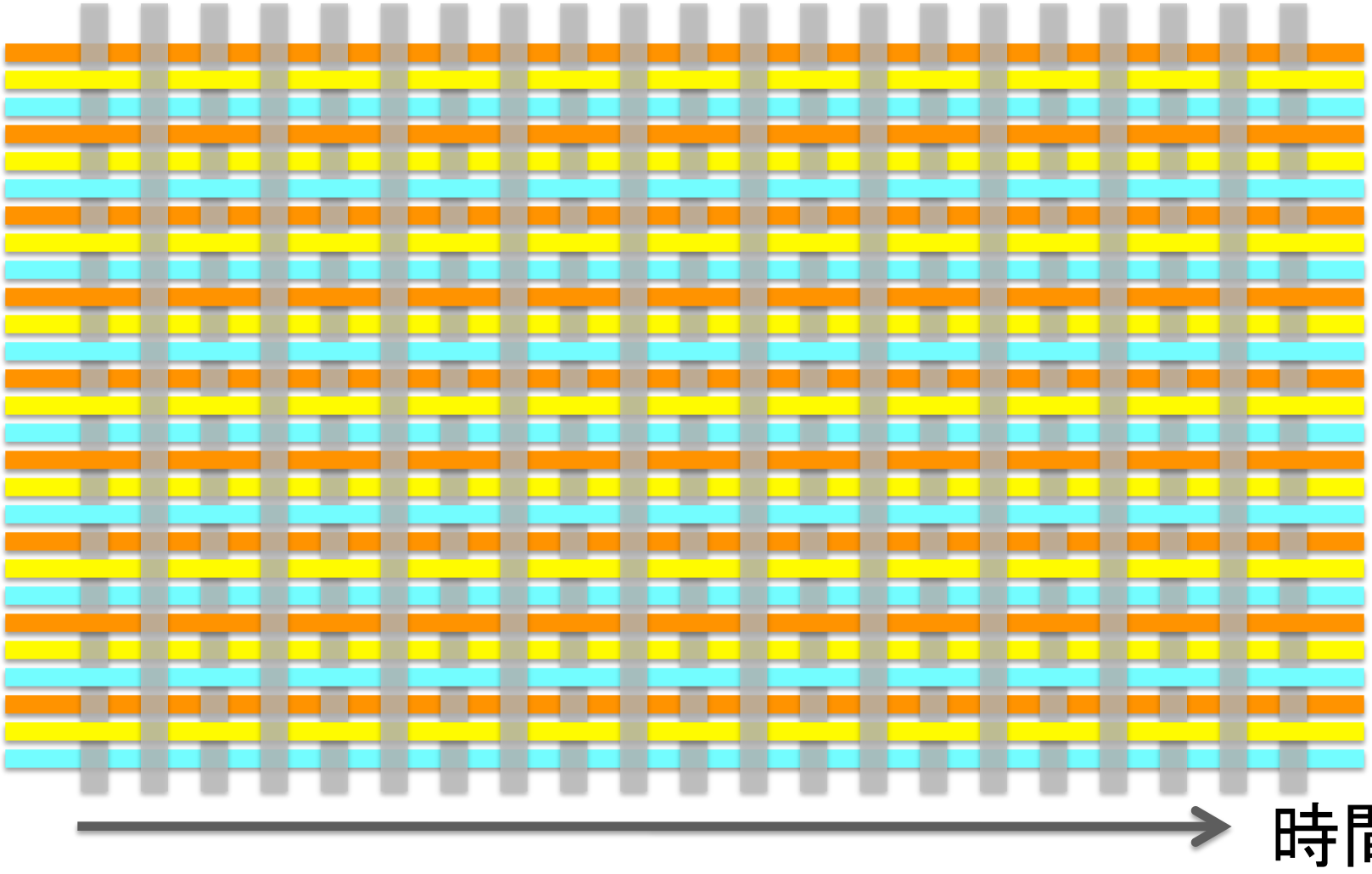

仕様化
実現
テスト
時間

並行プロセス

「｜」は、並行オペレータ


仕様化
実現
テスト
時間

プロセス ＝ 仕様化 ＋ 実現 ＋ テスト
× プロダクト



ビジネスプロセスと開発プロセスの連動

狭義のアジャイルプロセスは、仕様化、実現、テストを並行化したもの。
広義のアジャイルプロセスは、上記の他に、ビジネスプロセス、保守プロセスなど、あらゆる事項が並行化されます。

・・・

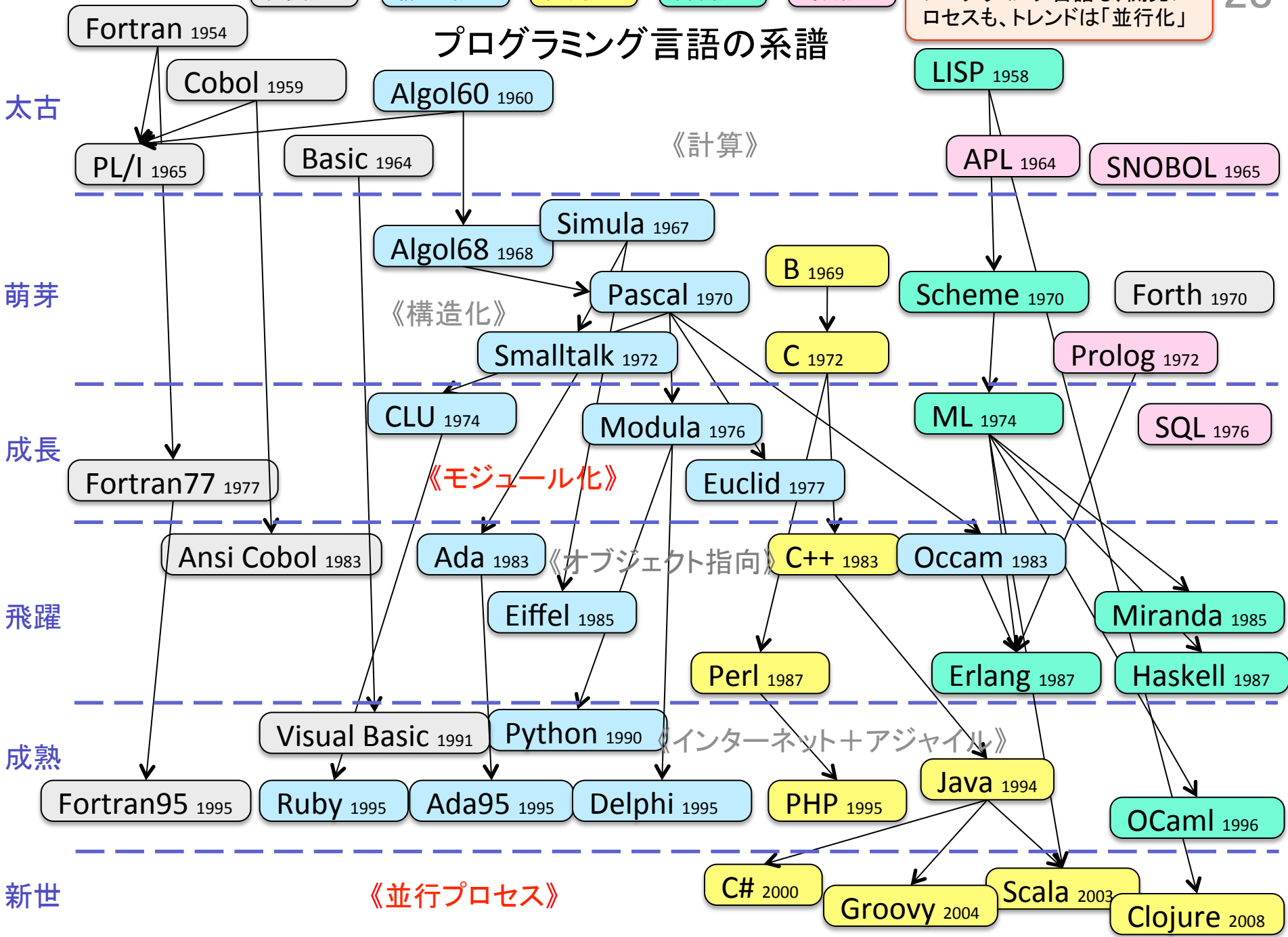
知の織物

＝ソフトウェア

プログラミング言語の系譜

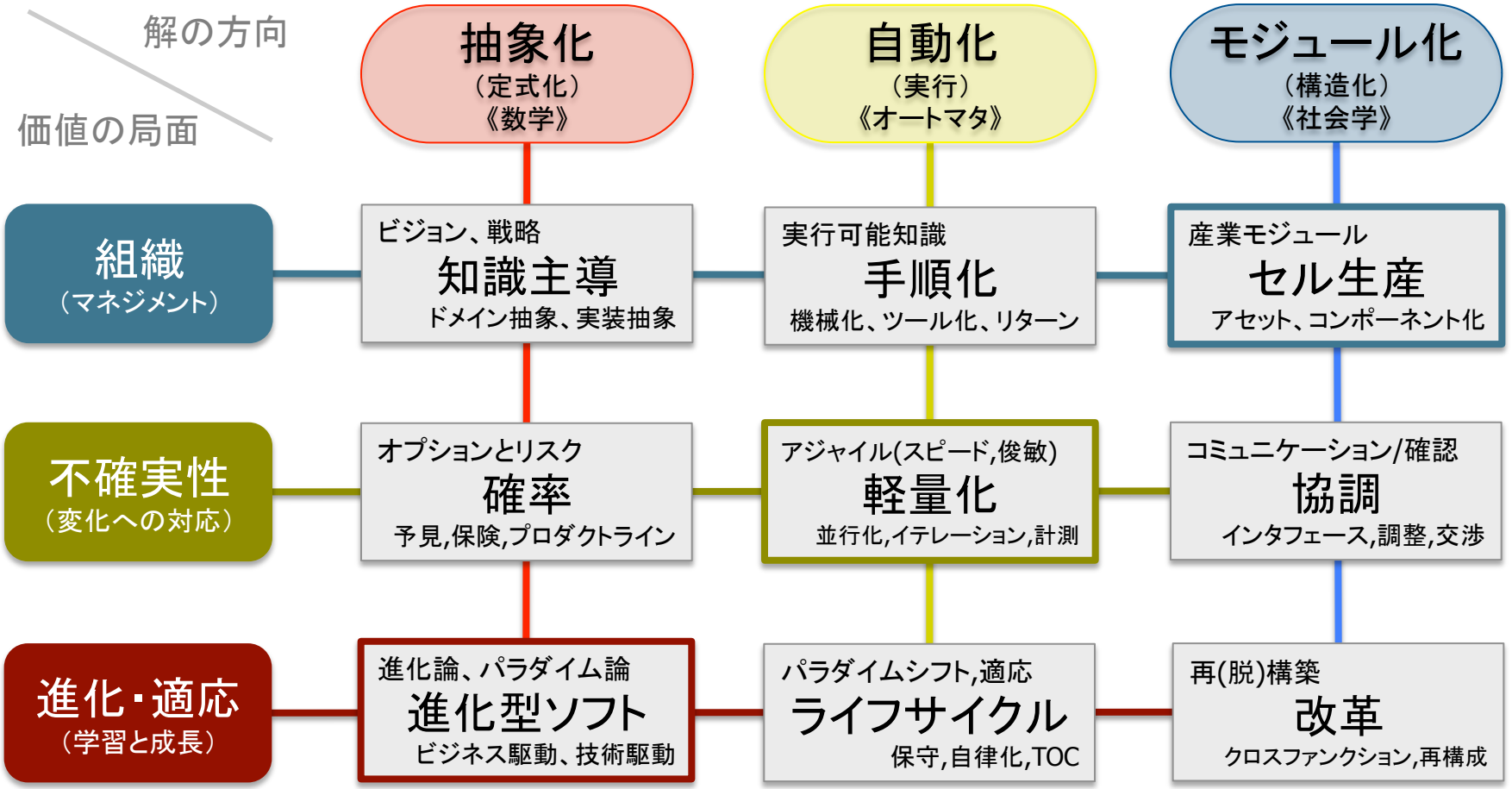
実務指向　構造化指向　実践指向　関数指向　専門指向

プログラミング言語も、開発プロセスも、トレンドは「並行化」



ソフトウェアが生み出す価値とそれを生み出す方法(解)との関係

価値と解のマトリクス



ベースラインとしての組織(マネジメント)に始まり、
アジャイルプロセスでは不確実性・変化への対応、
そして、次に、進化・適応の方法に向かっています。

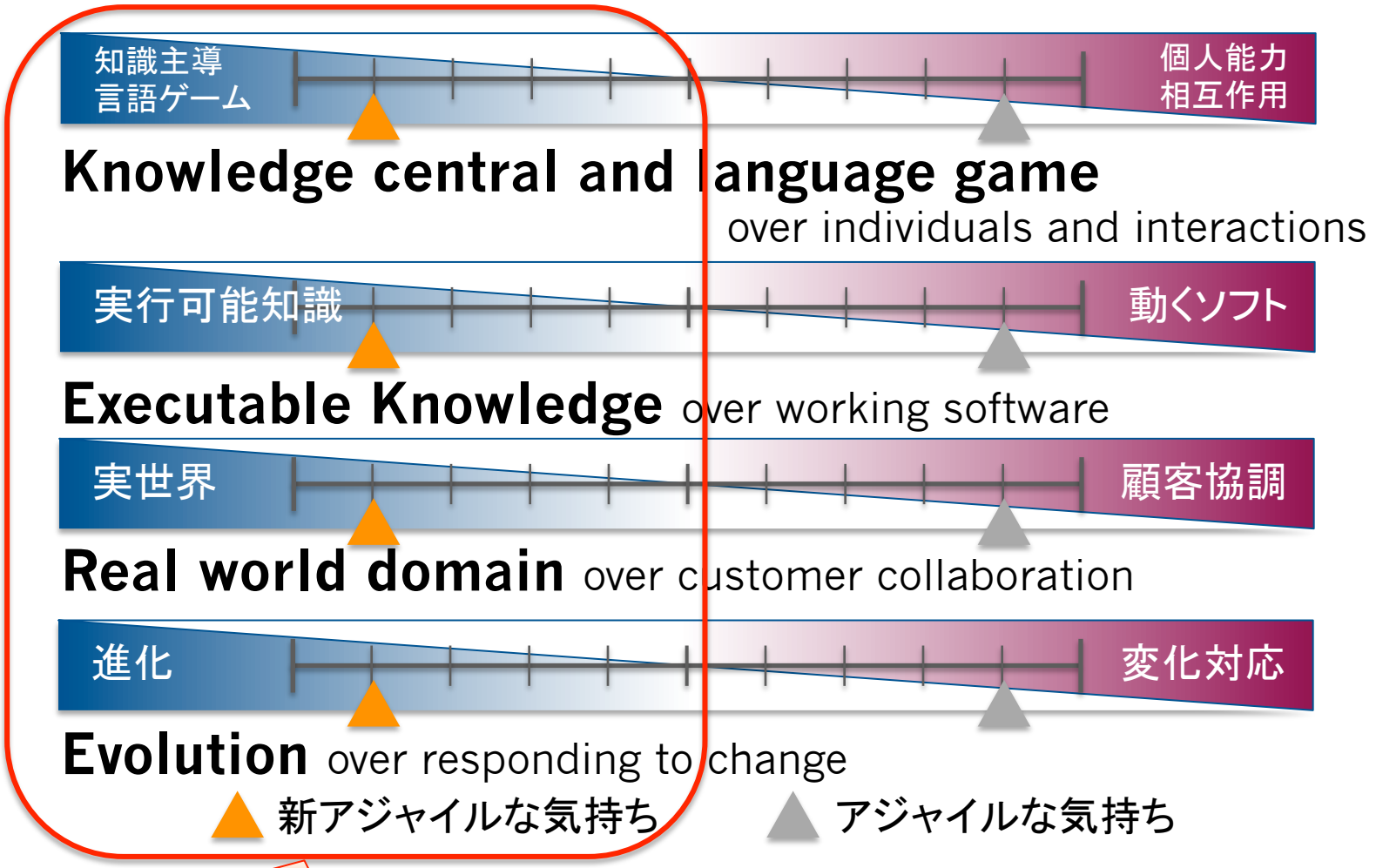
個人能力や相互作用	＞	プロセスやツール
動くソフト	＞	文書
顧客協調	＞	契約･交渉
変化対応	＞	計画遂行

アジャイルソフトウェア開発マニフェストの中の、より価値を置く事項を、もう少々掘り下げてみると、『アジャイルプロセスの未来』が見えてきます。

個人能力とは？　相互作用の本当の意味？　動くソフトとは？
ソフトウェアとは何か？　真の顧客協調とは？
変化はなぜ起こるのか？　・・・

新アジャイルプロセス宣言

New Manifesto for Agile S/W Development



今後の計画

開発・保守・利用の統合プロセス、実行可能知識の進化、
次世代のアーキテクチャ、分析手法、・・・