

「構成」という観点で考える

ソフトウェア開発

最近「構成」というキーワードが  
気になってきました。

いろんな物事を上手く回すためには  
構成を考えるのが大事なんやなーと  
感じます。

構成という観点を軸に、私の目指す  
ソフトウェア開発のあり方を考えて  
みます。

# 構成、構成力

- ・ いろいろな要素を組みあわせて全体として意味を作りあげる。
- ・ 一部だけに注力すると全体ではちぐはぐになる。  
全体しか見ないと具体性に欠ける。
- ・ 良い構成を作るには、俯瞰視点と細部の注目の両方を行き来して。  
全体がうまく回るように設計しないといけない。

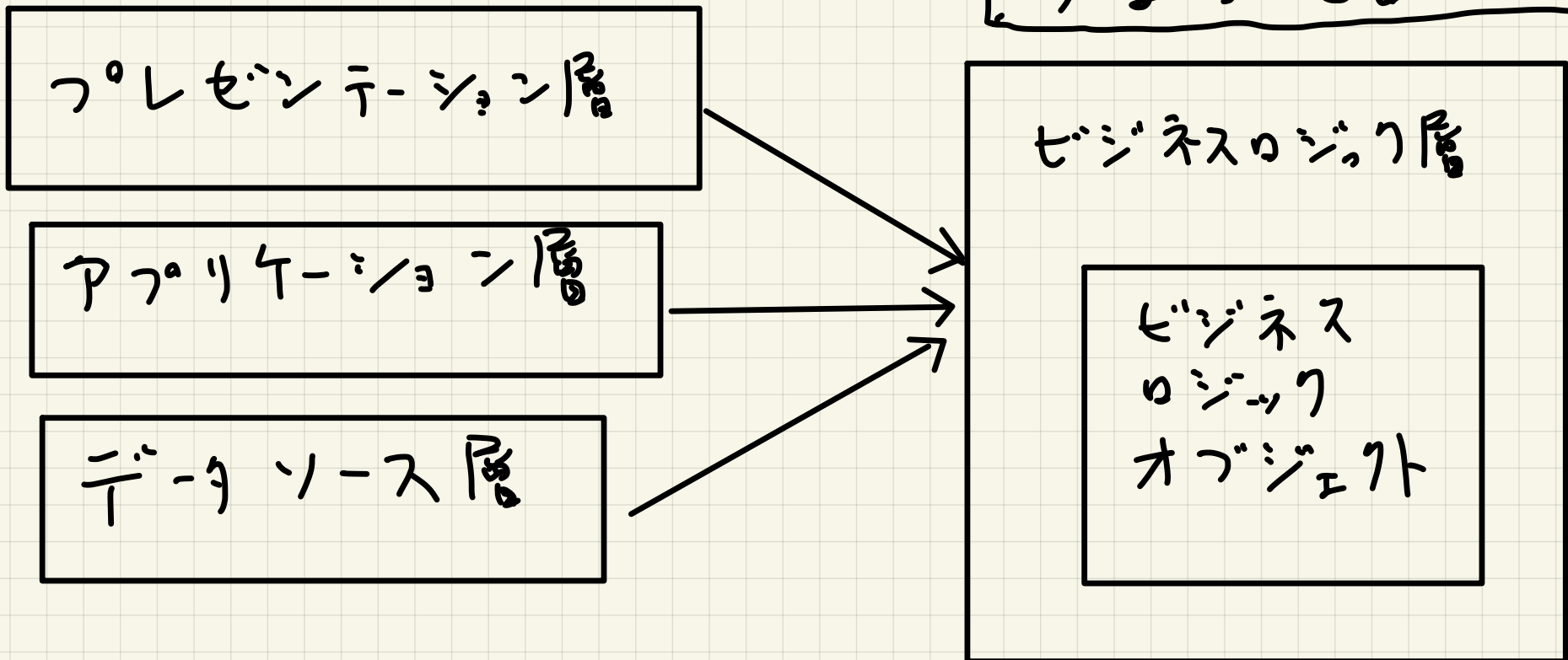
いくつかのケースで構成  
の例を見ていきます。

- ・ クラス構成
- ・ サービス の構成
- ・ プロジェクト計画の構成
- ・ 継続的な組織学習の構成

# クラス構成

- ・ ビジネスロジックオブジェクトを中々に
- ・ プレゼン層などの都合で設計がゆがまないように。

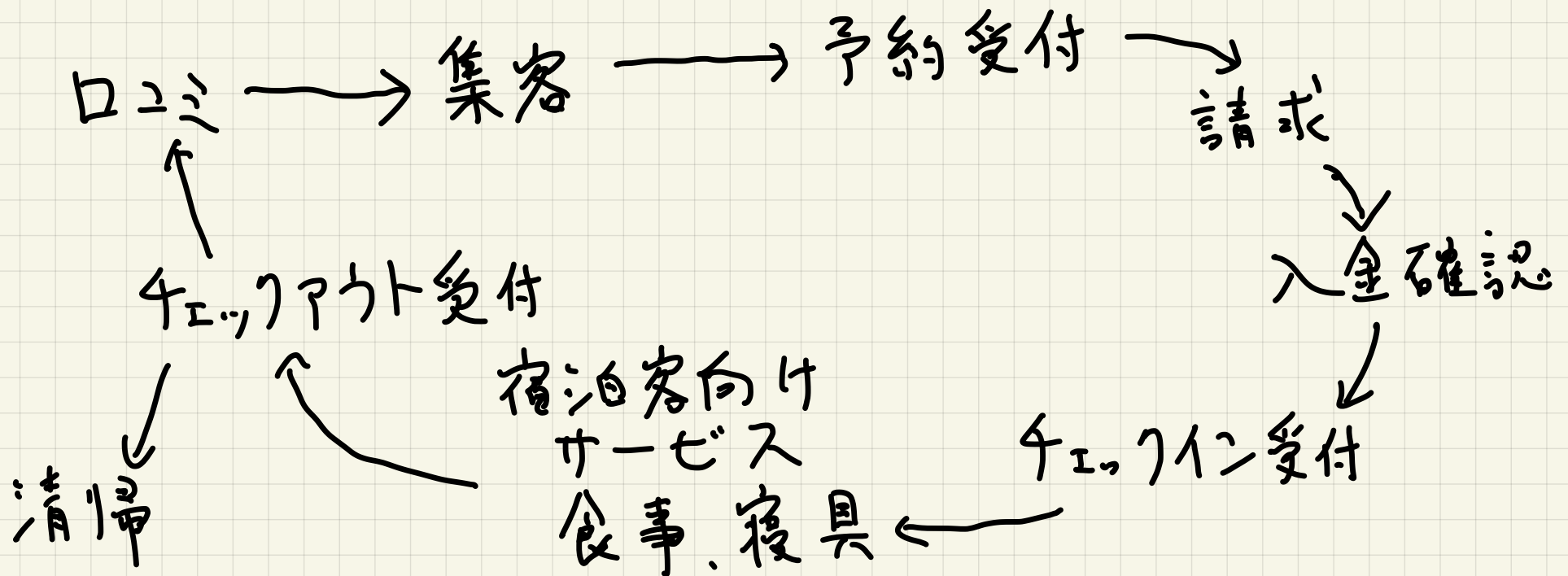
スケールする設計につながる。



# サービスの構成

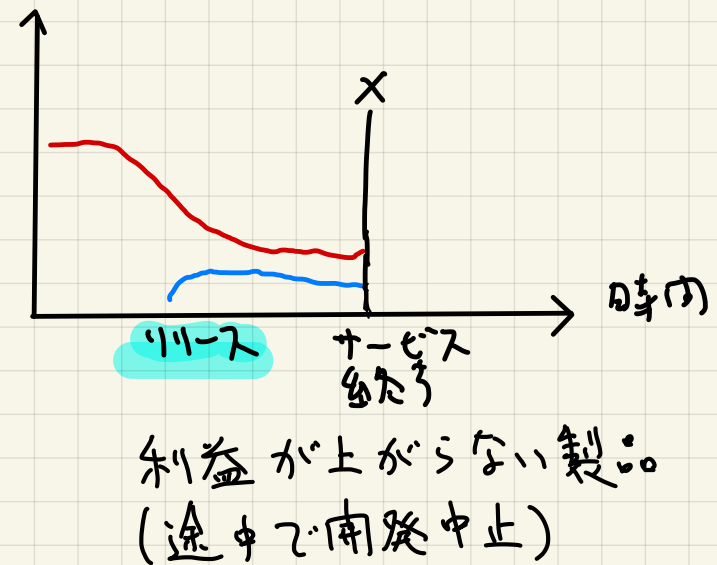
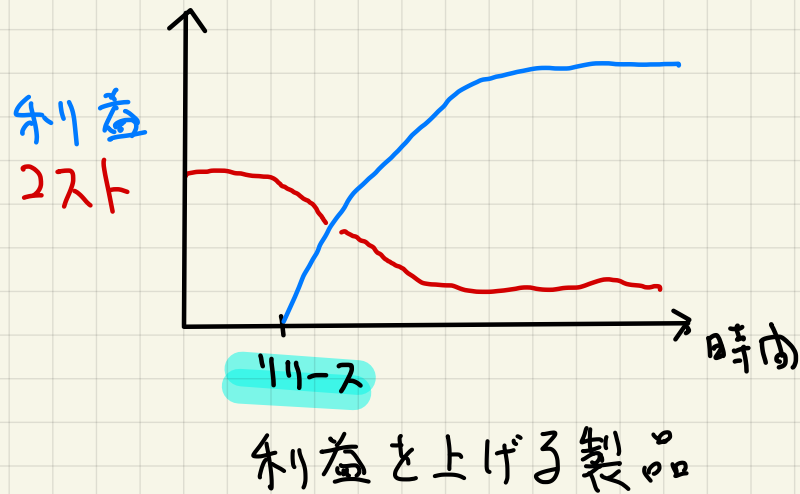
- ・コア業務を軸に、関連業務とのつながりもふまえて全体としてビジネスがうまくまわるように設計する。

## 観光旅館ビジネス



# プロジェクト計画の構成(コスト計画)

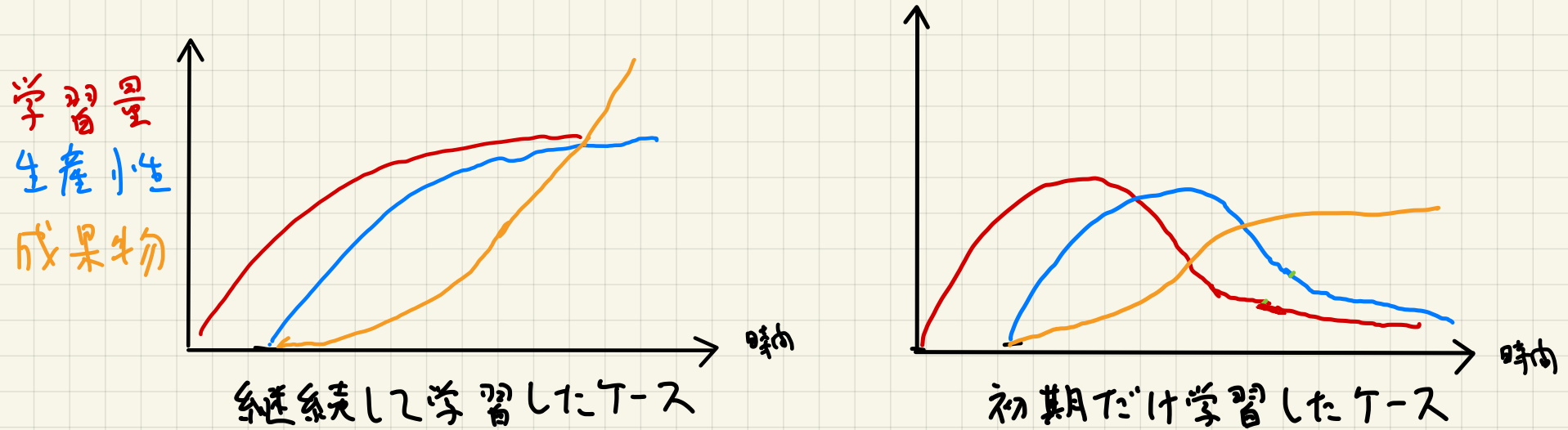
- ・ 開発対象に需要があり、利益が上がるか、  
実際リリースしないと確実にわからない。



- ・ やる価値のある開発か早期に判断
- ・ フィードバックを受けて設計を改善(プログラムの発展性)  
↑ 早期のフィードバックを計画に組み込む

# プロジェクト計画の構成 (学習と生産性)

ビジネス、技術の学習が生産性に影響する。  
ただし生産性、成果が出るのにタイムラグがある。



学習量に注目することで生産性を安定させる

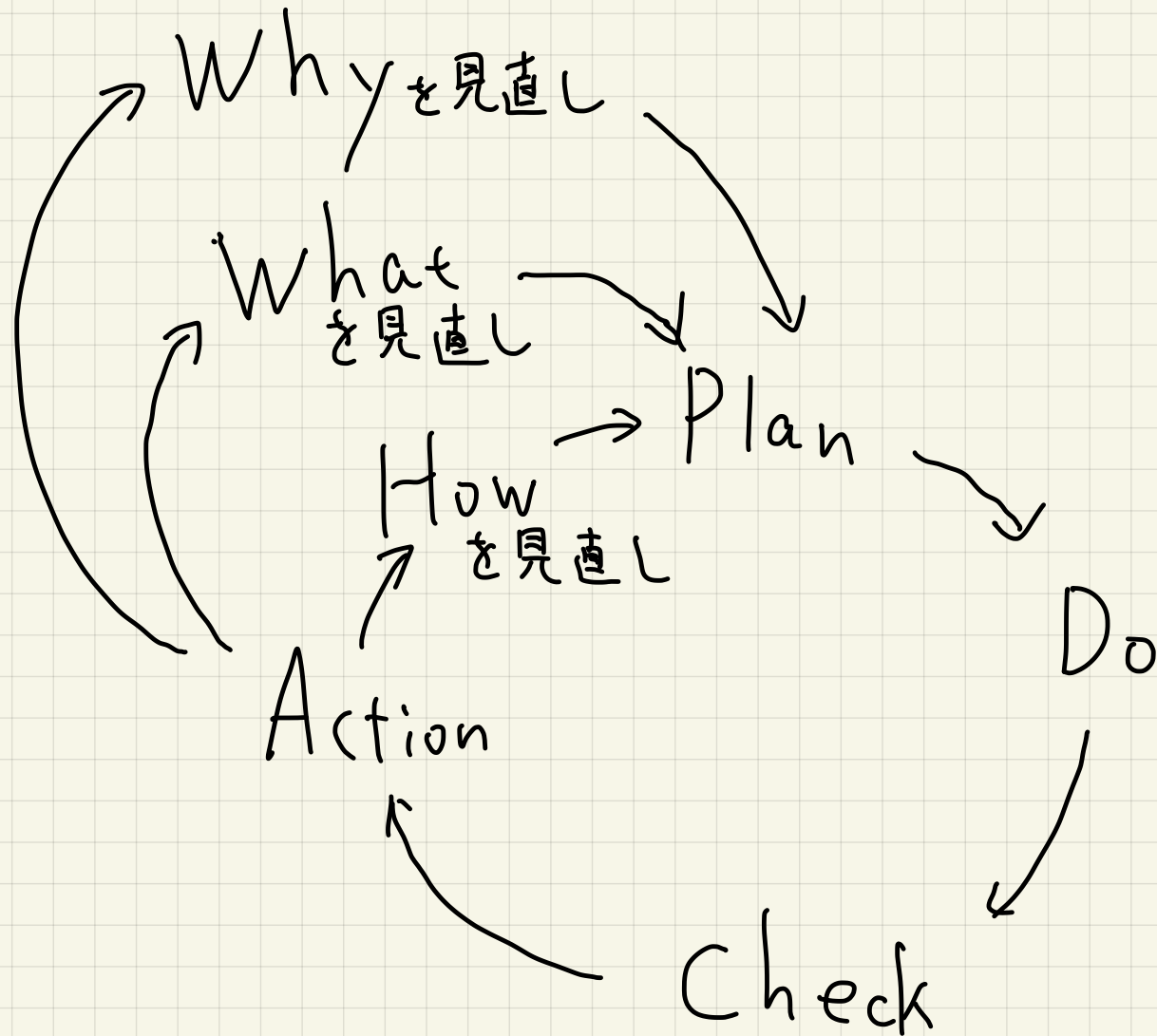
↑ 継続的に学習時間を確保

↑ 開発に実験を組み込む (体験知)



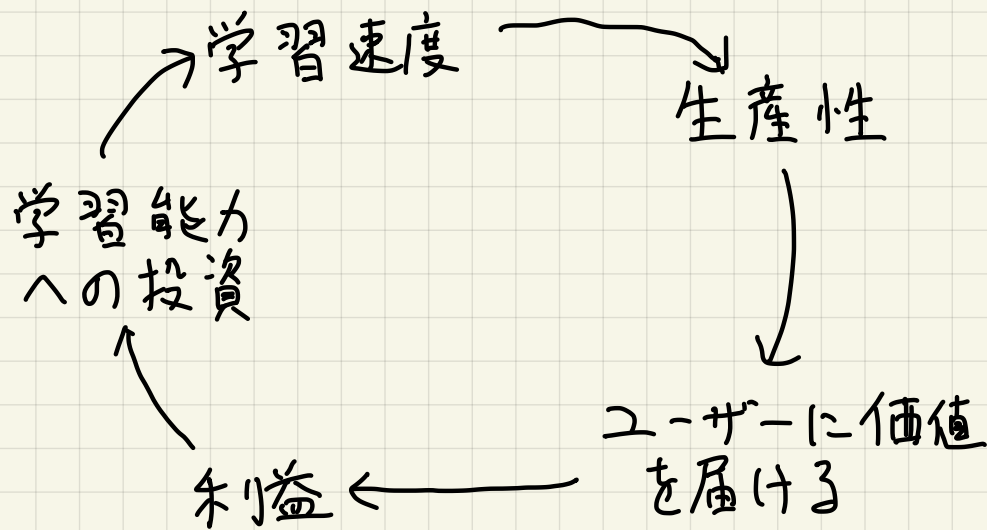
# 効果的な学習計画の構成

Howを見直すシングルループ学習だけでなく、  
ダブルループ学習、トリプルループ学習を利用  
して、What、Whyも定期的に見直す。

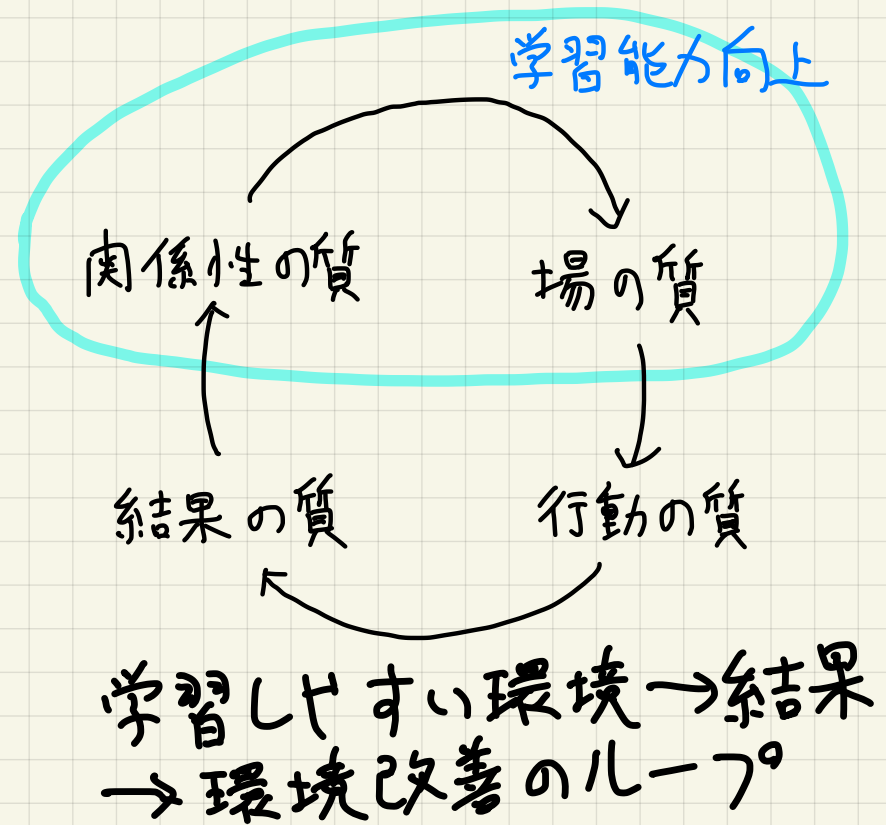


# 継続的な組織学習の構成

- ・ソフトウェア開発組織において、最重要の能力は学習能力。
- ・継続的に学習能力を成長するため、ビジネスに学習する環境を組み込む。



利益を組織学習に  
還元するループ



まとめ:

いろいろな場面で構成力が求められる。

- ・ グラス構成
- ・ サービスの構成
- ・ プロジェクト計画の構成
- ・ 継続的な組織学習の構成

経験知を増やして

構成力を伸ばしたい。