embarcadero Developer Camp

RAD Studio で実践する
継続的インテグレーション

アプリとデベロッパーの価値を拡張するエッセンス

長沢 智治
テクニカル エバンジェリスト
アトラシアン株式会社
re-workstyle.com　@tomohn

アプリとデベロッパーの価値

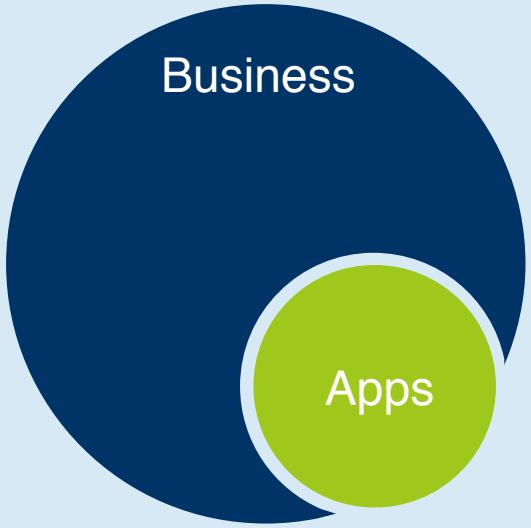
ビジネスとアプリケーションの進化

90s

C/S

- ✓コード品質
- ✓開発者中心
- ✓分業

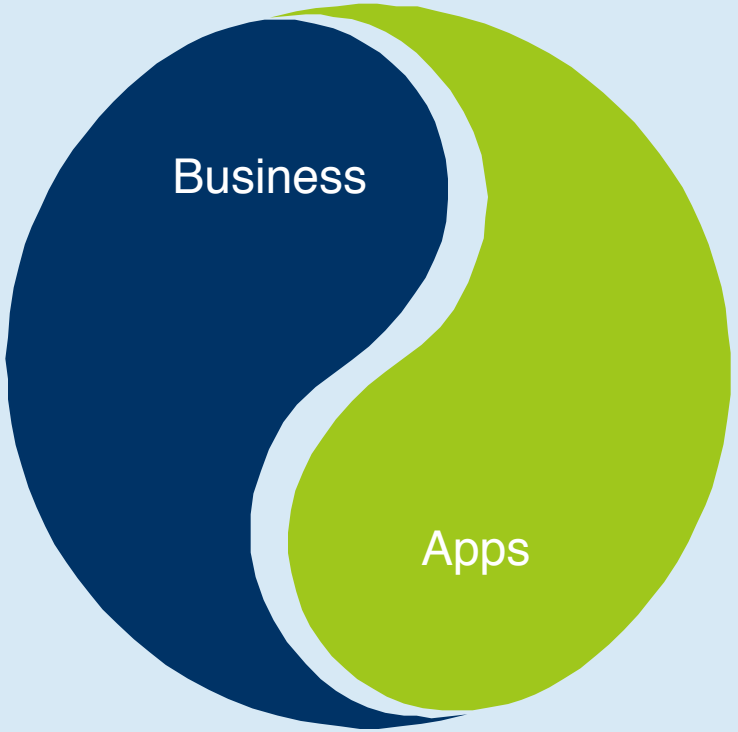
00s



Web サービス

- ✓サービス品質
- ✓開発チーム中心
- ✓分業から協調

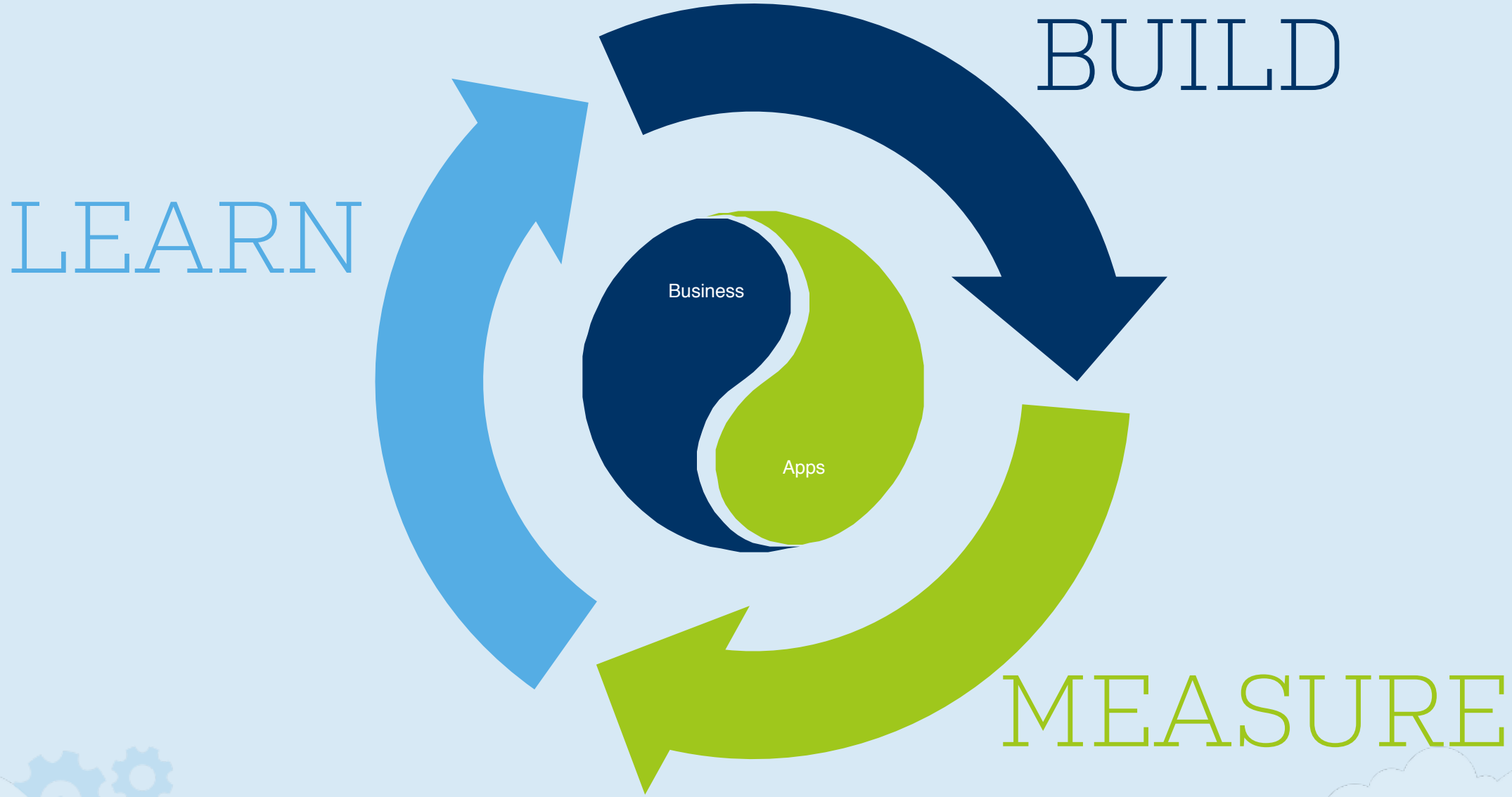
10s



マルチデバイス + サービス

- ✓ビジネス品質
- ✓開発と運用
- ✓協調

ビジネスを駆動するアプリケーションへ



創造 | 成果 | 変革

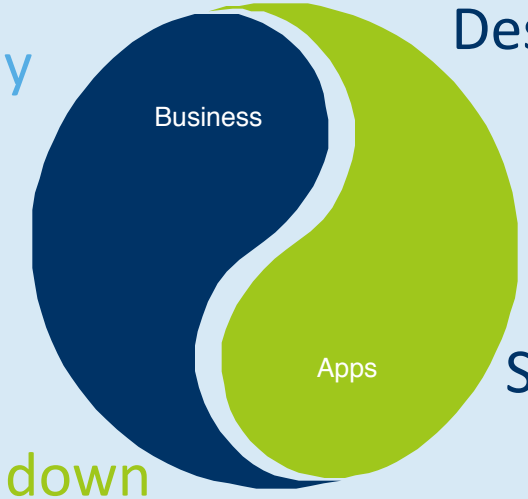
ビジネスを駆動するプラクティス

Feedback loop Deployment

継続的デリバリー

Small Batch Production Ready

Acceptance Test DVCS



Design for Operations

DevOps

MTTR Customer needs

Service Desk Cloud Cycle Time

Retrospective

Scrum Burn down

DoD Value Up

Task Board Agile Time Box

Backlog eXtreme Programing

Pair Programming CI TDD

Continuous Integration

継続的インテグレーション

ウィキペディア

継続的インテグレーション

CI: Continuous Integration

アプリケーション作成時の
品質改善や納期の短縮のための習慣のことです。

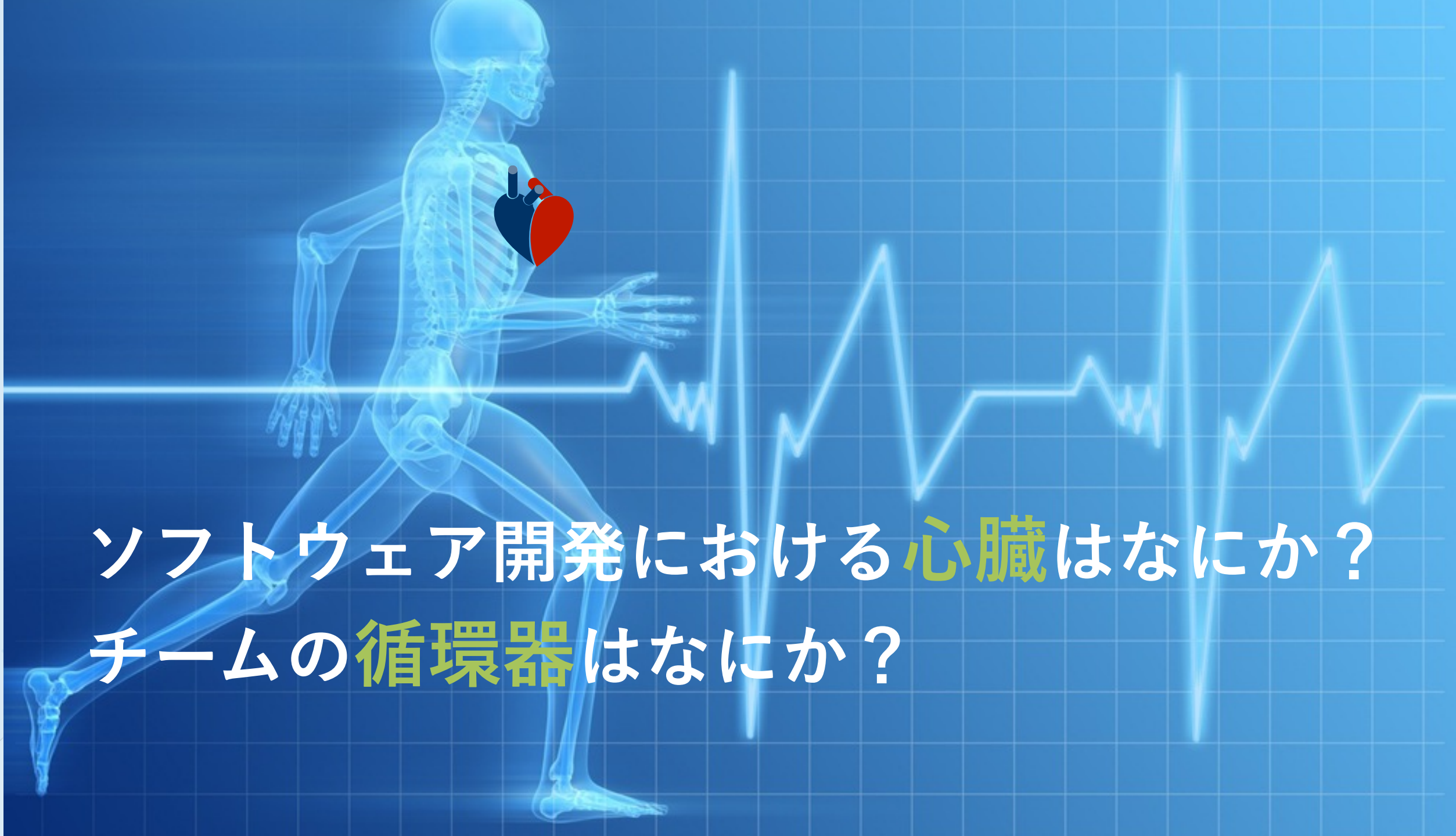
XP のプラクティスの一つでビルドやテスト、
インスペクションなどを継続的に実行していく
ことを意味する。

品質改善

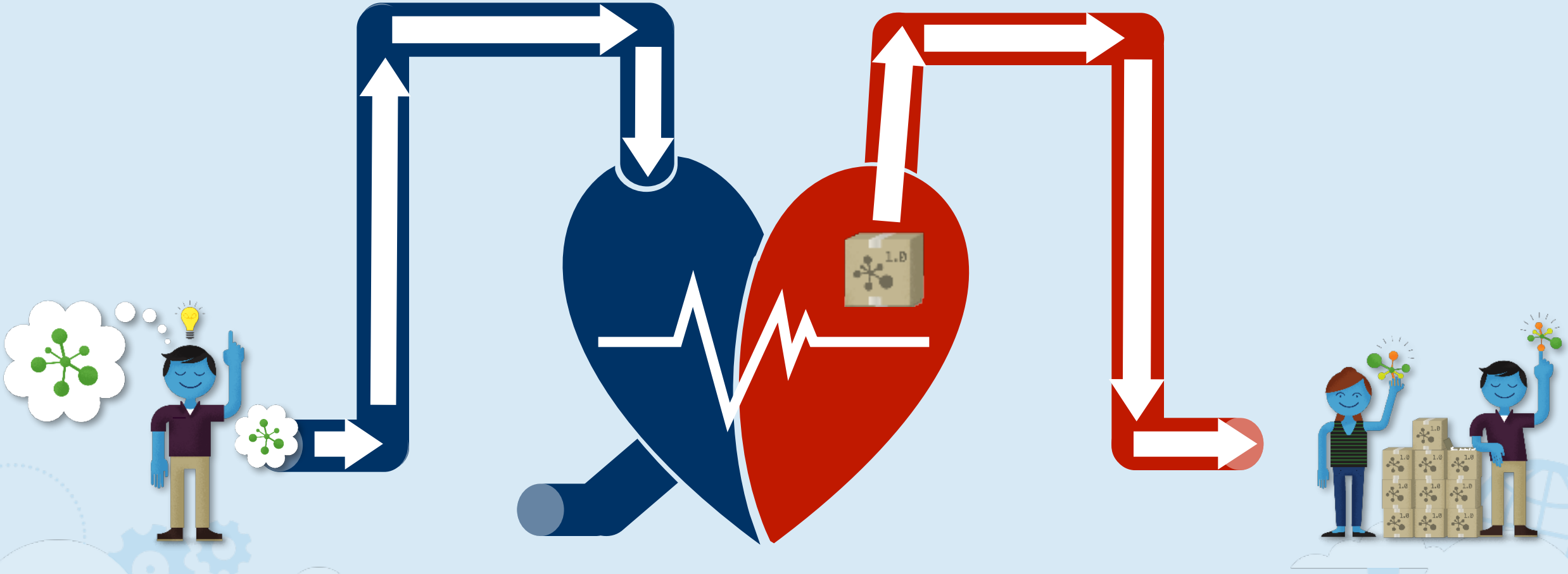
納期の短縮

継続的
ビルド

習慣

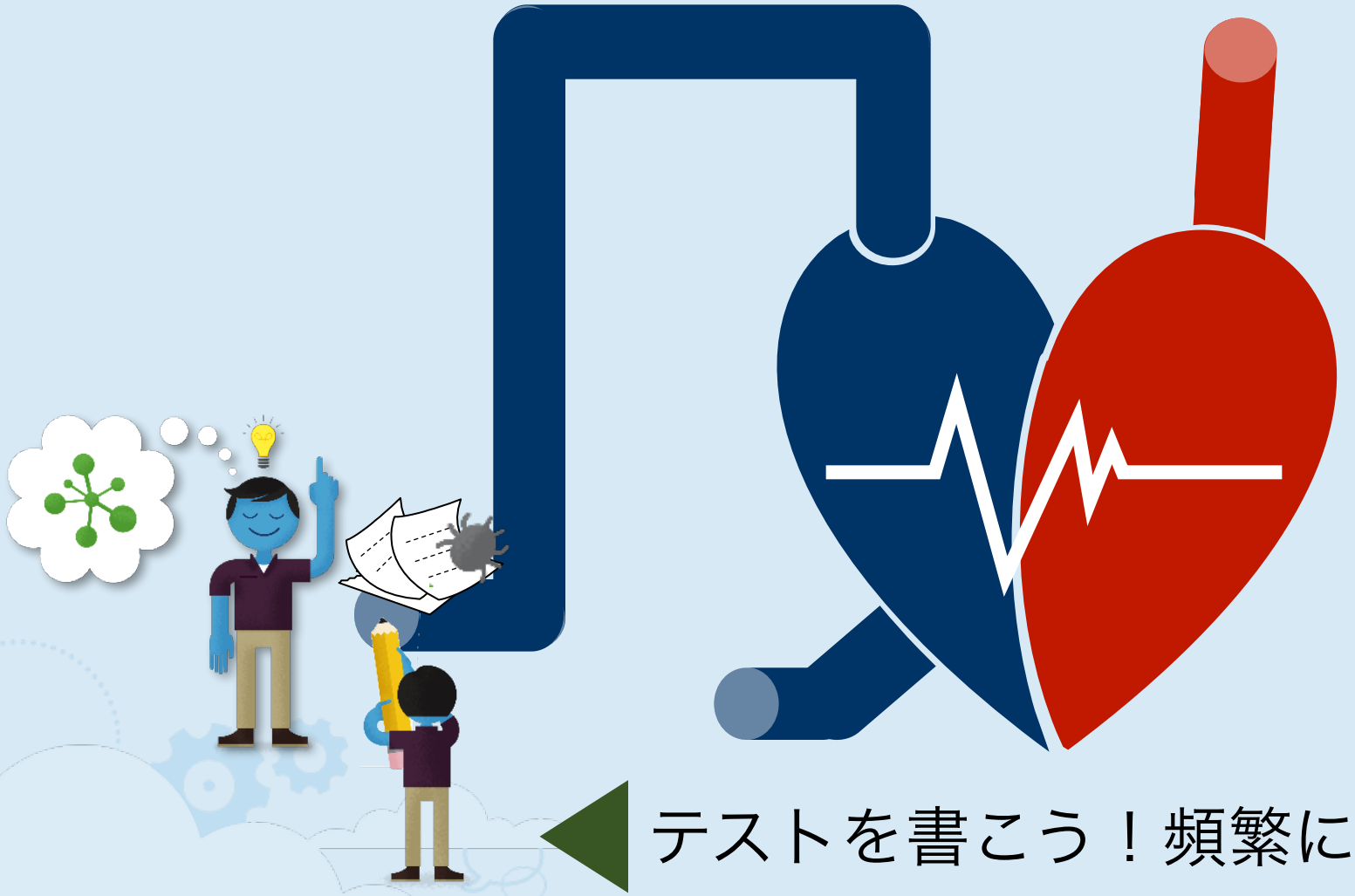

ソフトウェア開発における心臓はなにか？
チームの循環器はなにか？

ソフトウェア開発 ～ アイデアを価値に転換する
Software Delivery



開発とビルドのリズムとコスト

Develop Build



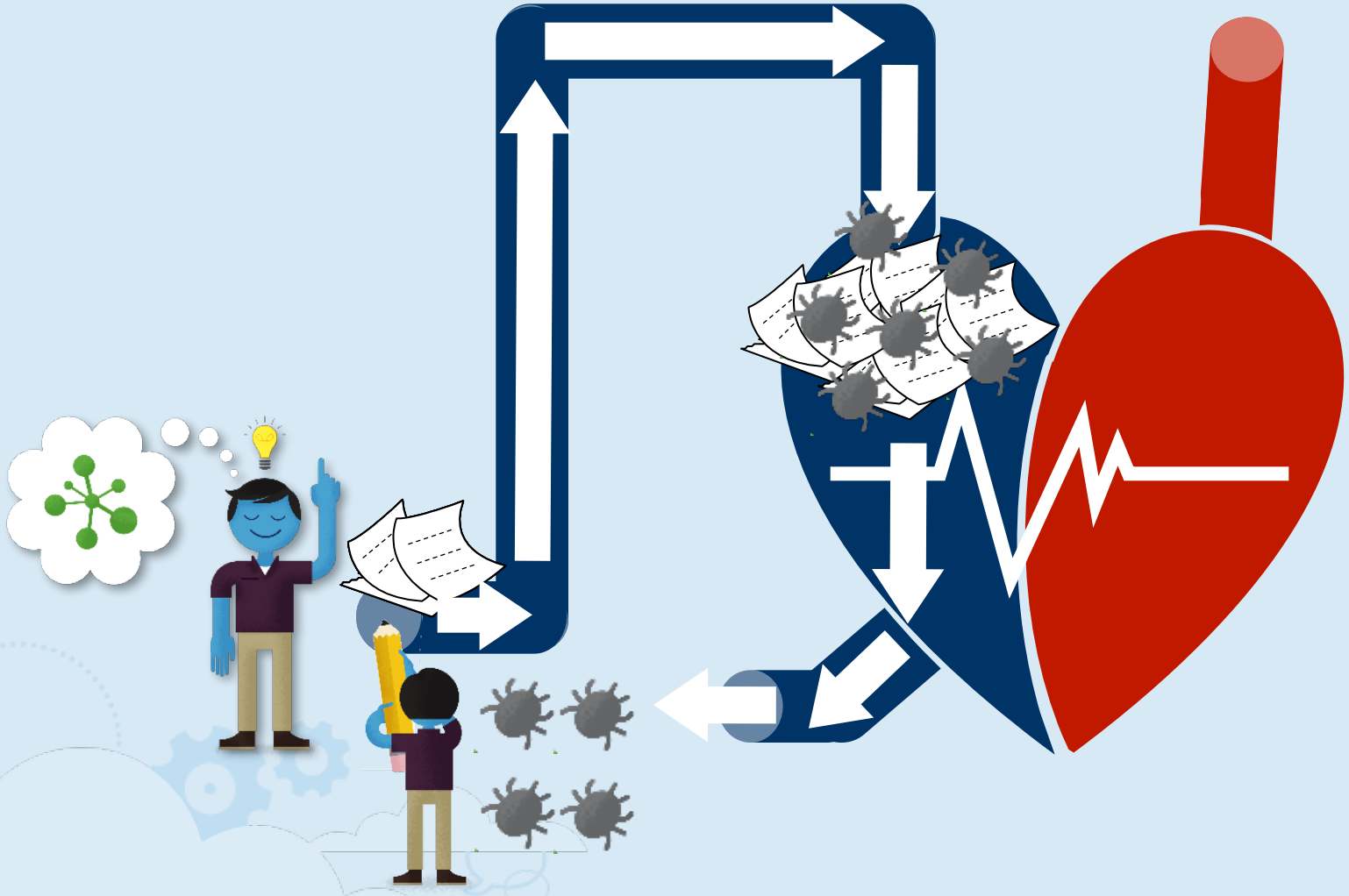
1 : 10 : 100 の法則

- ✓ 開発: コスト × 1
- ✓ ビルド: コスト × 10
- ✓ テスト: コスト × 100

テストを書こう！頻繁にコミットしよう！ (DVCS)

開発とビルドのリズムとコストの蓄積

Develop Build

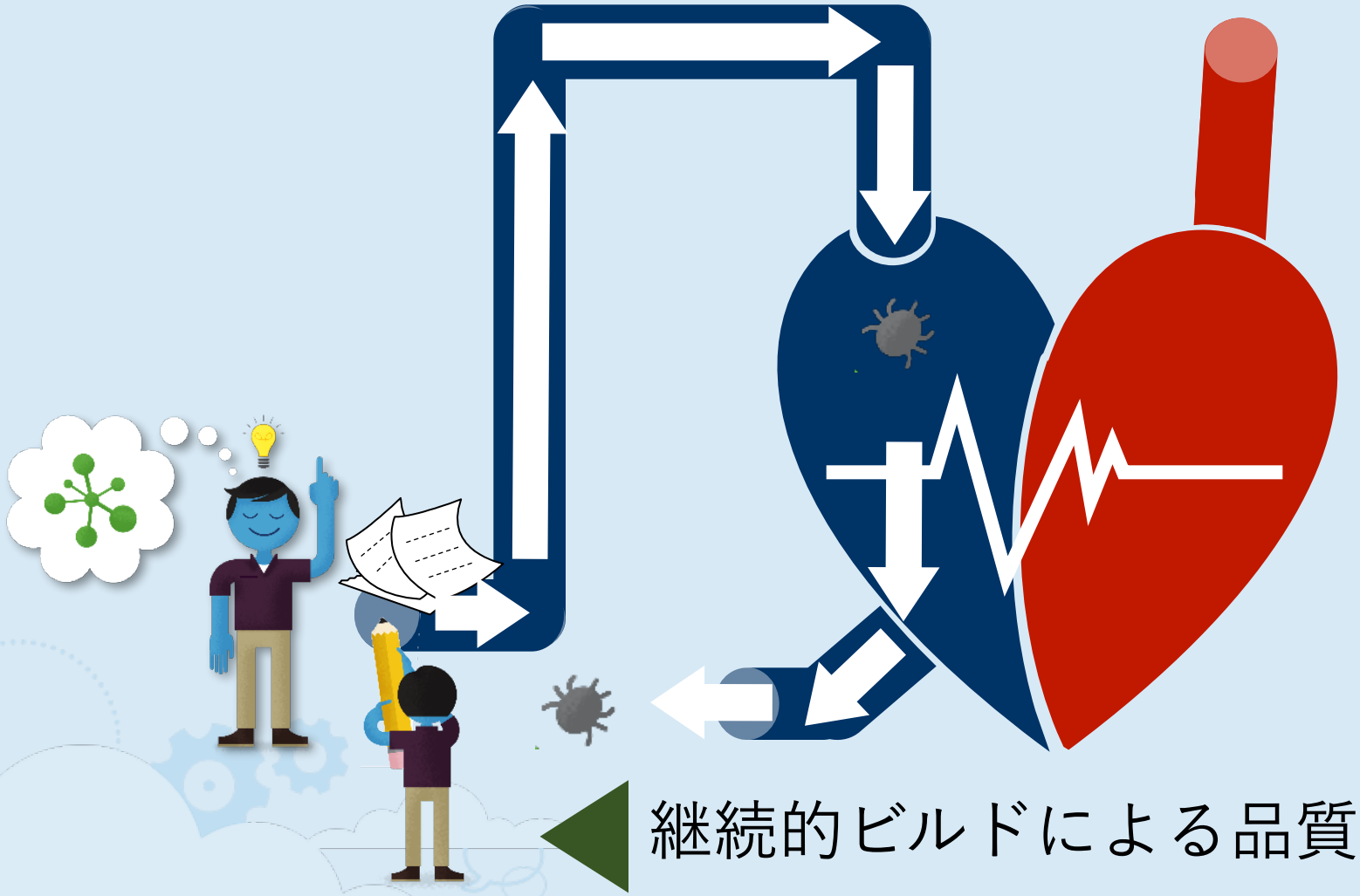


1 : 10 : 100 の法則

- ✓ 開発: コスト × 1
- ✓ **ビルド: コスト × 10**
- ✓ テスト: コスト × 100

開発とビルドのリズムとコストの改善

Develop Build



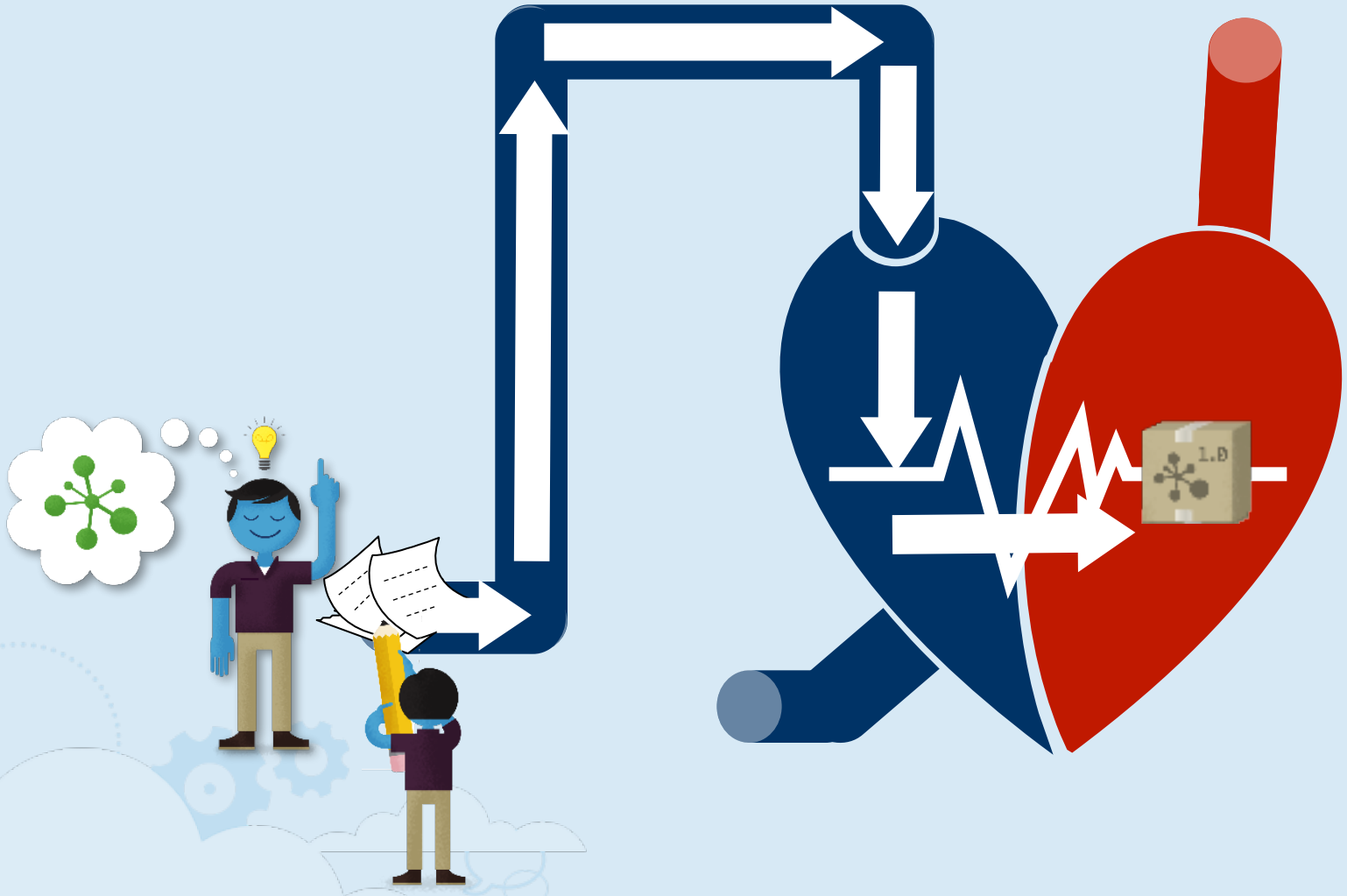
1 : 10 : 100 の法則

- ✓ 開発: コスト × 1
- ✓ **ビルド: コスト × ~~1~~ 10**
- ✓ テスト: コスト × ~~10~~ 100

継続的ビルドによる品質改善とデリバリーの短縮

開発とビルドのリズムからデプロイのリズムへ

Develop Build Deploy

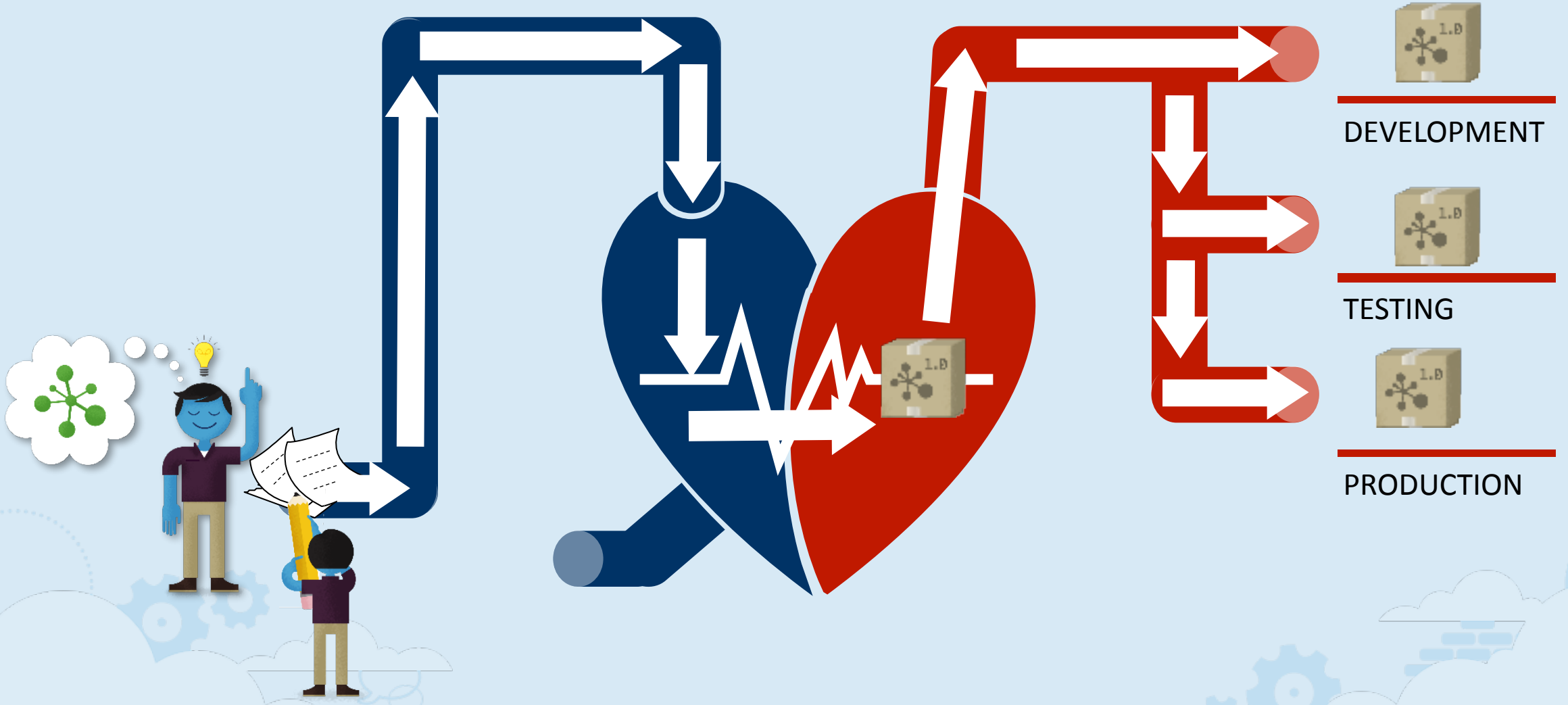


1 : 10 : 100 の法則

- ✓ 開発: コスト × 1
- ✓ **ビルド: コスト × 10**
- ✓ テスト: コスト × 100
- ✓ **デプロイ: コスト ×100**

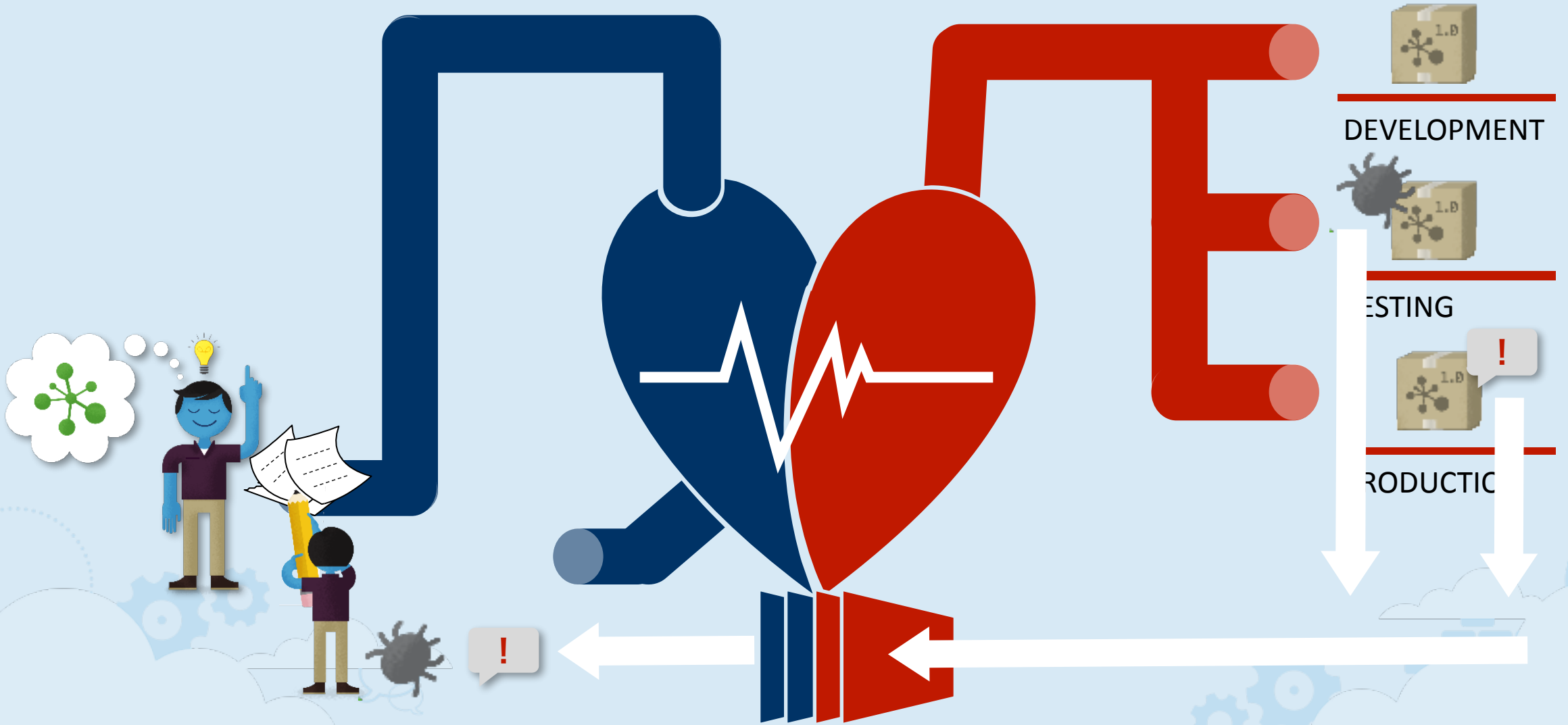
開発とビルドのリズムからデプロイのリズムへ

Develop Build Deploy



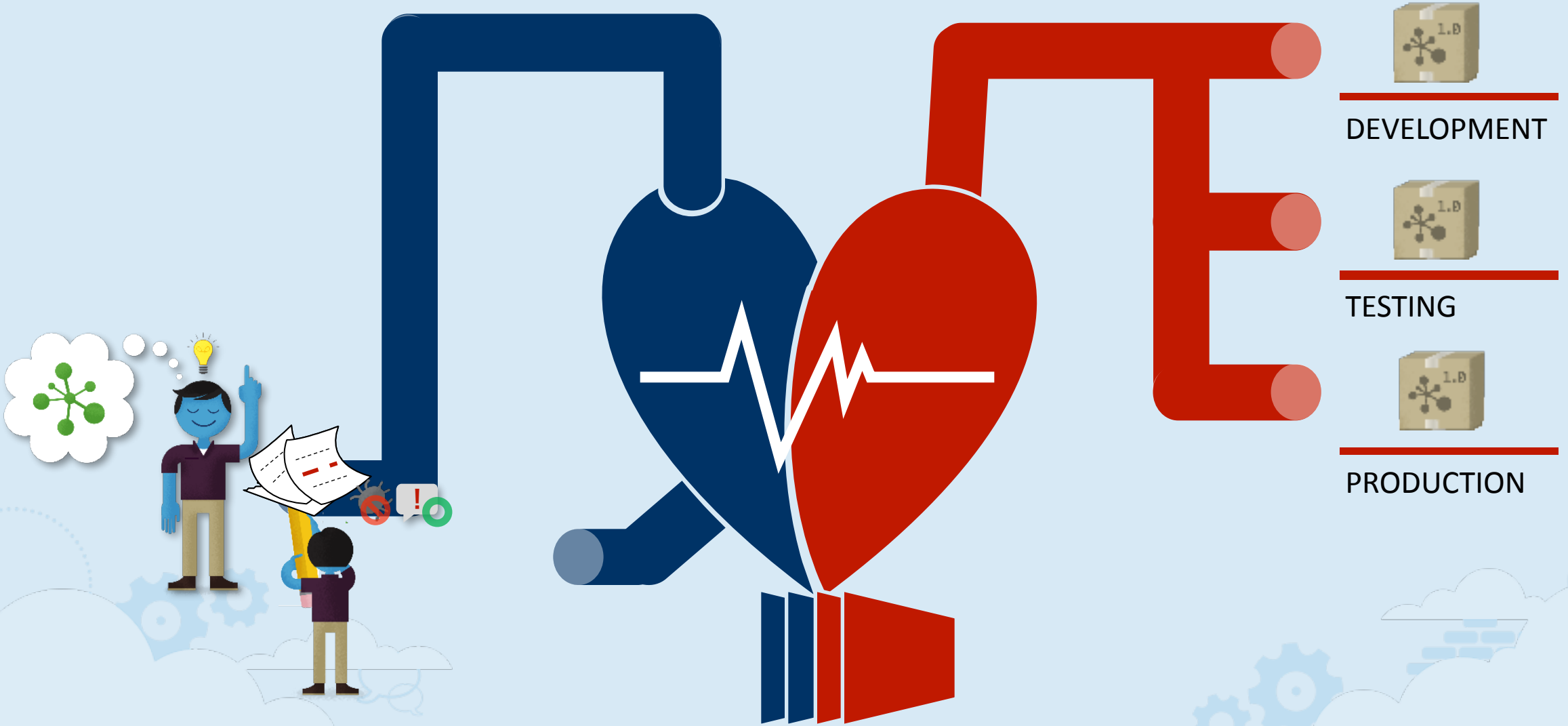
開発とビルドのリズムからデプロイのリズムへ

Develop Build Deploy



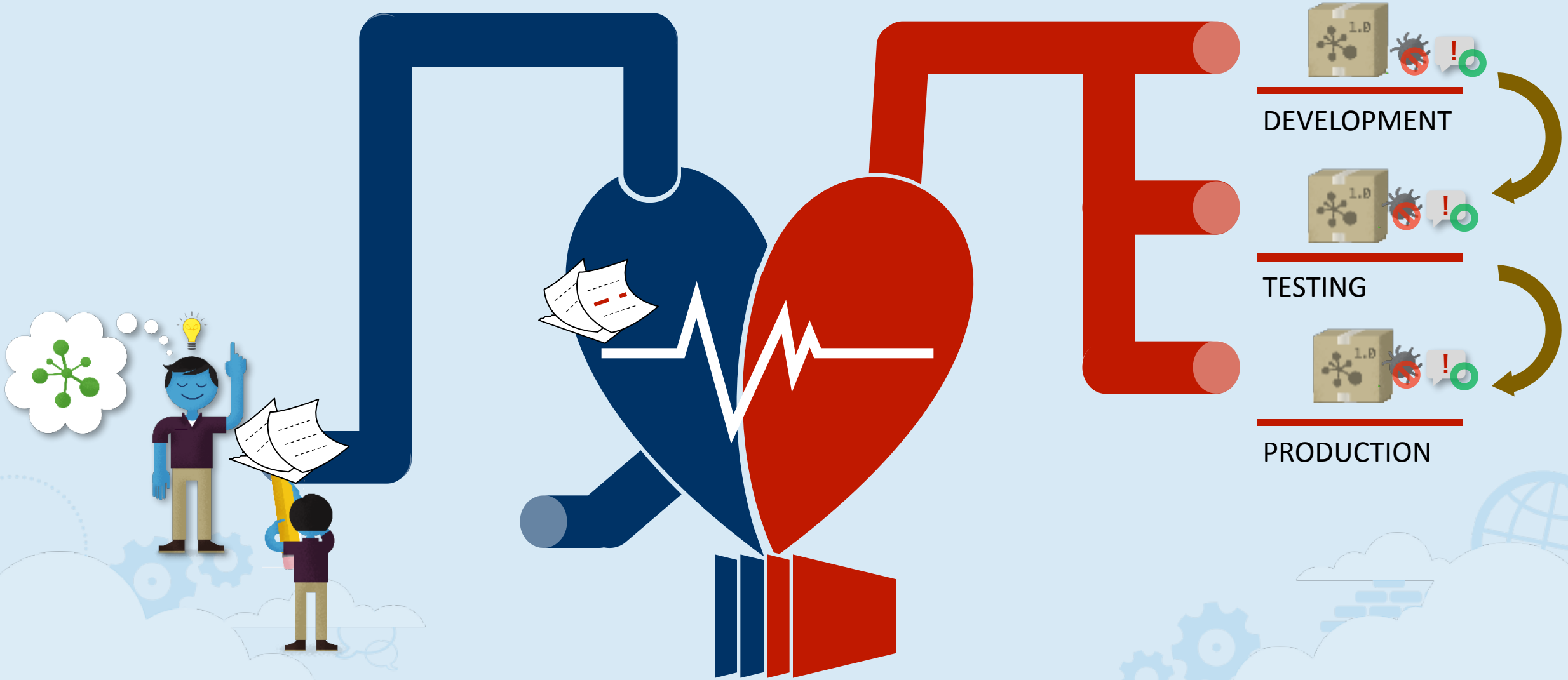
開発とビルドのリズムからデプロイのリズムへ

Develop Build Deploy



開発とビルドのリズムからデプロイのリズムへ

Develop Build Deploy



開発者のコード変更は直接デリバリーへ

Develop Build Deploy

継続的インテグレーション
VCS + CI (+BTS/ITS)

アイデアとバグを
コードに転換

手を抜けないコード
の品質の作りこみ

DEVELOPMENT

TESTING

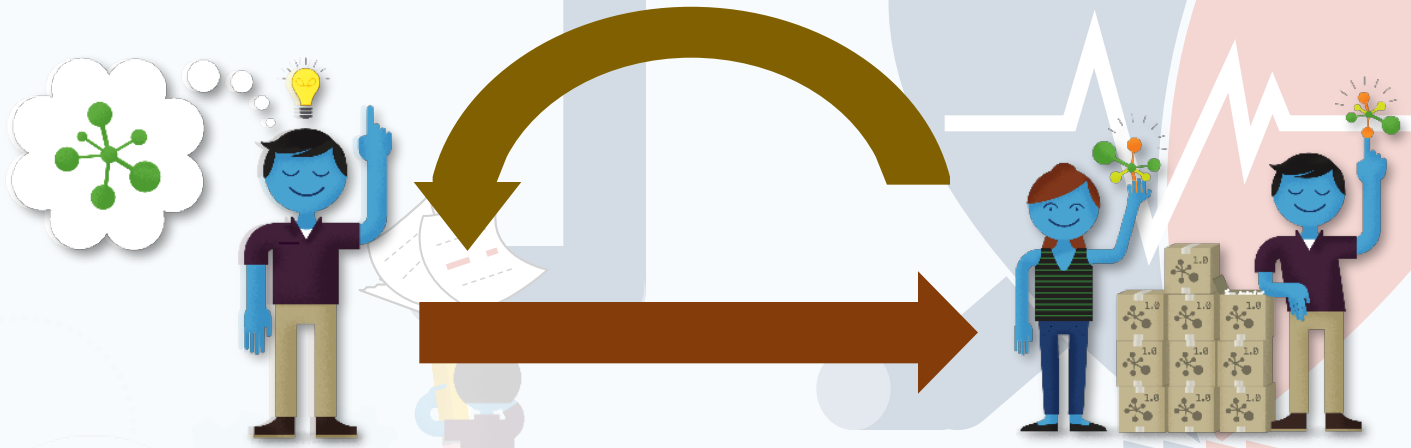
PRODUCTION

デプロイの自動化
リリース管理

フィードバックの
収集と適切な反映

開発者のコード変更は直接デリバリーへ

Develop Build Deploy



1 : 10 : 100 の法則

- ✓ 開発: コスト × 1
- ✓ ビルド: コスト × ~~1~~ 10
- ✓ テスト: コスト × ~~5~~ 100
- ✓ デプロイ: コスト × ~~10~~ 100

開発者とアプリ、要求とコードとビルドをつなぐ

Develop Build Deploy

開発者同士:

ITS/BTS — VCS — CI

チーム同士:

開発者と、
- デザイナー
- テスター
- マネージャー

ITS/BTS

利害関係者:

開発チームと、
- 企画
- 運用
- 顧客

ITS/BTS — DEPLOY

開発者とアプリ、要求とコードとビルドをつなぐ

Develop Build Deploy

開発者同士:

ITS/BTS — VCS — CI

チーム同士:

開発者と、
- デザイナー
- テスター
- マネージャー

ITS/BTS

利害関係者:

開発チームと、
- 企画
- 運用
- 顧客

ITS/BTS — DEPLOY

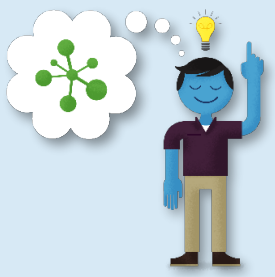
アトラシアン製品にみる
開発支援ツールの進化とカバー範囲

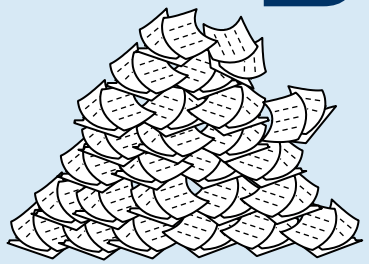
Atlassian

embarcadero
Developer Camp

開発者とアプリ、要求とコードとビルドをつなぐ

Develop Build Deploy





開発者同士:

チーム同士:
開発者と、
- デザイナー
- テスター
- マネージャー

利害関係者:
開発チームと、
- 企画
- 運用
- 顧客

SourceTree
DVCS クライアント
無償のデファクトスタンダード

Bitbucket Stash
DVCS リポジトリ
コードレビューや連携可能なリポジトリ管理

Bamboo
継続的インテグレーション
技術依存しない自動ビルドツール
ビルド管理

継続的デプロイメント
自動デプロイとデプロイ状況の管理

JIRA
BST / ITS: 要求、バグ、タスクの追跡, 変更管理
ドキュメントやソースコードの変更要素 (Issues) の追跡と管理
各種の成果物の粒度の調整と各成果物をつなぐ重要な役割
開発者とアプリの価値をわかりやすく示すのに欠かせない

Confluence
ライブ ドキュメント共有
企画や仕様書を陳腐化させない。ドキュメントから協調、思考と経験の形式知化

HipChat
チーム内外のコミュニケーション インフラ
タイムラインでアクティビティを通知、対処の円滑化

RAD Studio + Atlassian 製品での継続的インテグレーション

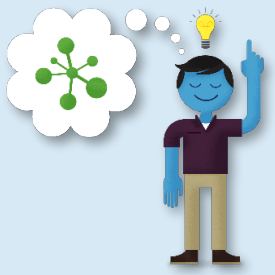
Demo

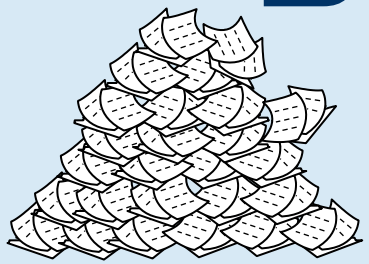
Atlassian

embarcadero Developer Camp

開発者とアプリ、要求とコードとビルドをつなぐ

Develop Build Deploy





開発者同士:

チーム同士:
開発者と、
- デザイナー
- テスター
- マネージャー

利害関係者:
開発チームと、
- 企画
- 運用
- 顧客

SourceTree
DVCS クライアント
無償のデファクトスタンダード

Bitbucket Stash
DVCS リポジトリ
コードレビューや連携可能なリポジトリ管理

Bamboo
継続的インテグレーション
技術依存しない自動ビルドツール
ビルド管理

継続的デプロイメント
自動デプロイとデプロイ状況の管理

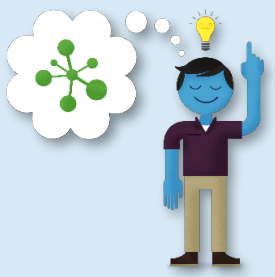
JIRA
BST / ITS: 要求、バグ、タスクの追跡, 変更管理
ドキュメントやソースコードの変更要素 (Issues) の追跡と管理
各種の成果物の粒度の調整と各成果物をつなぐ重要な役割
開発者とアプリの価値をわかりやすく示すのに欠かせない

Confluence
ライブ ドキュメント共有
企画や仕様書を陳腐化させない。ドキュメントから協調、思考と経験の形式知化

HipChat
チーム内外のコミュニケーション インフラ
タイムラインでアクティビティを通知、対処の円滑化

開発者とアプリ、要求とコードとビ… つなぐ

Develop　　Buil…



開発者同士:

チーム同士:
開発者と、
- デザイナー
- テスター
- マネージャー

利害関係者:
開発チームと、
- 企画
- 運用
- 顧客

D
無

D
コー
なり

JIRA
BST / ITS
ドキュメント
各種の成果物の
開発者とアプリの

Confluence
ライブ ドキュメント共有
企画や仕様書を陳腐化させない。ト

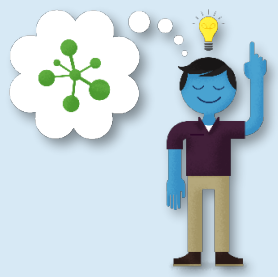
HipChat
チーム内外のコミュニケーション イ
タイムラインでアクティビティを通知、対処の円滑化

RAD Studio (Delphi, C++Builder) でのテスト

- ✓ テスト容易性の高い設計と実装
  - ✓ ビューとロジックの分離 (MVC, MVVM)
  - ✓ デバッグ実行 ≠ テスト
- ✓ テスティング フレームワークの活用
  - ✓ DUnit
  - ✓ DUnitX
- ✓ DUnit
  - ✓ DUnit: xUnit 互換の Delphi ユニットテスト
    - ✓ RAD Studio で標準搭載
    - ✓ IDE からのテスト実行に特化
      - ✓ GUI とコマンドライン実行
      - ✓ テスト結果は対話的 (ログなし)

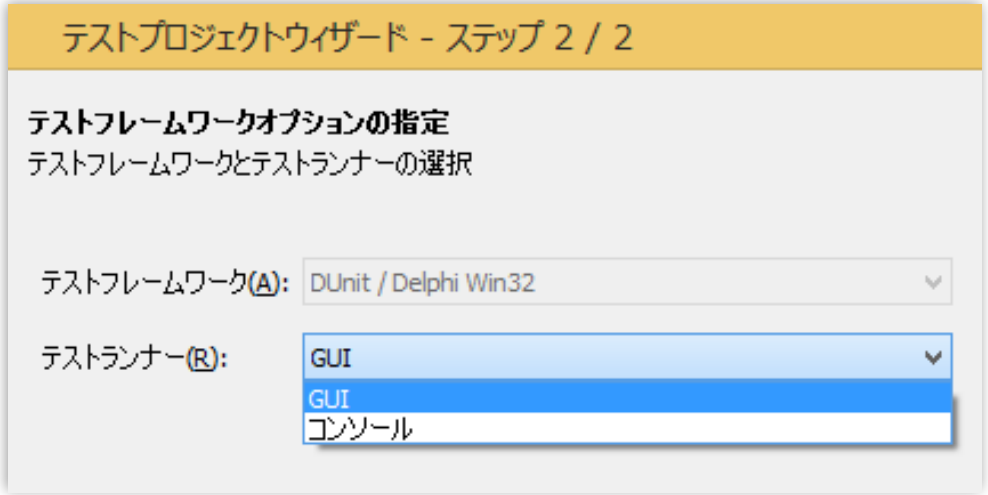
開発者とアプリ、要求とコードとビ… つなぐ

Develop　　Buil…



DUnit でのテストの作成

ウィザードによるテストの作成



開発者同士:

チーム同士:
開発者と、
- デザイナー
- テスター
- マネージャー

利害関係者:
開発チームと、
- 企画
- 運用
- 顧客

JIRA

BST / ITS
ドキュメント
各種の成果物の
開発者とアプリの

Confluence

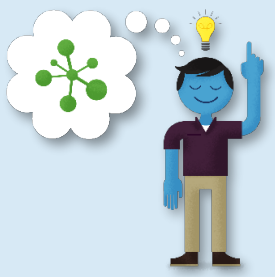
ライブ ドキュメント共有
企画や仕様書を陳腐化させない。ド

HipChat

チーム内外のコミュニケーション イ
タイムラインでアクティビティを通知、対処の円滑化

開発者とアプリ、要求とコードとビ… つなぐ

Develop　　Bui…



開発者同士:

チーム同士:
開発者と、
- デザイナー
- テスター
- マネージャー

利害関係者:
開発チームと、
- 企画
- 運用
- 顧客

JIRA
BST / ITS
ドキュメント…
各種の成果物の…
開発者とアプリの…

Confluence
ライブ ドキュメント共有
企画や仕様書を陳腐化させない。ド…

HipChat
チーム内外のコミュニケーション イ…
タイムラインでアクティビティを通知、対処の円滑化…

DUnit でのテストロジックの記述

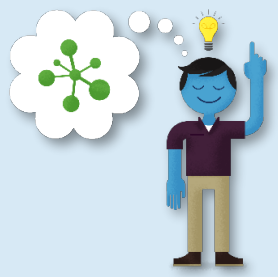
```
procedure TestTCalc.TestAdd_Case1;
  var
    _result : System.Integer;
    y: System.Integer;
    x: System.Integer;
  begin
    x := 1;
    y := 1;
    _result := aTCalc.Add(x, y);
    CheckEquals(x+y, _result);
  end;
```

テスト対象の実行

テスト対象の検証

開発者とアプリ、要求とコードとビ… つなぐ

Develop　　Buil…



開発者同士:

チーム同士:
開発者と、
- デザイナー
- テスター
- マネージャー

利害関係者:
開発チームと、
- 企画
- 運用
- 顧客

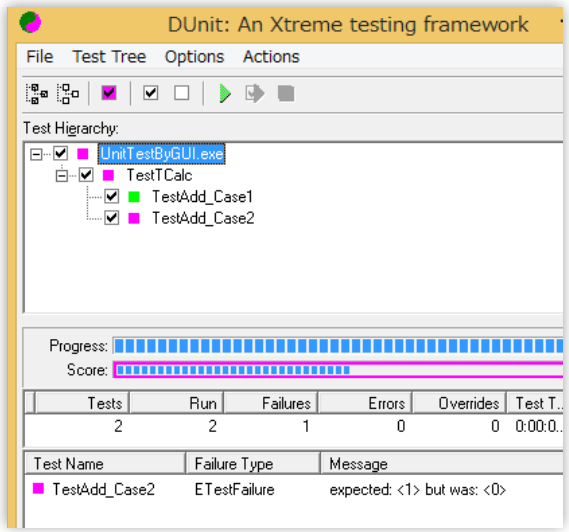
JIRA
BST / ITS
ドキュメント…
各種の成果物の…
開発者とアプリの…

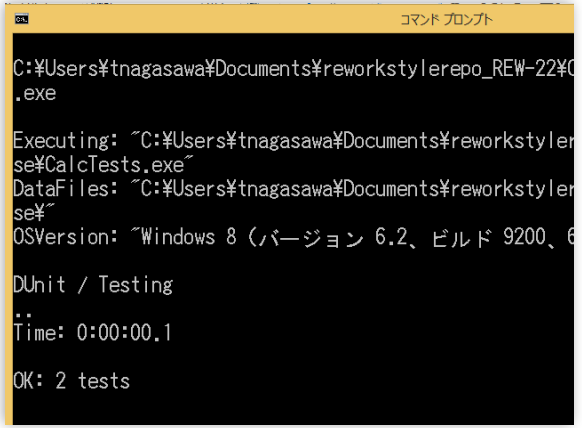
Confluence
ライブ ドキュメント共有
企画や仕様書を陳腐化させない。ト…

HipChat
チーム内外のコミュニケーション イ…
タイムラインでアクティビティを通知、対処の円滑化…

DUnit でのテストの実行

テストの実行

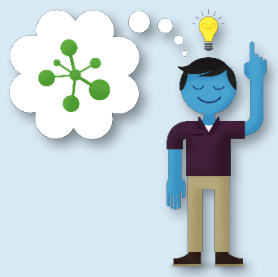




テスト結果がファイルに出力できない
＝ 継続的インテグレーションで検証困難！

開発者とアプリ、要求とコードとビルド

Develop Build



開発者同士:

チーム同士:
開発者と、
- デザイナー
- テスター
- マネージャー

利害関係者:
開発チームと、
- 企画
- 運用
- 顧客

JIRA

BST /
ドキュ
各種の成
開発者と

Confluence ライブ ドキュメ
企画や仕様書を陳腐

HipChat チーム内外のコミ
タイムラインでアクティ

DUnit でのテストロジックの記述

XMLTestRunner2.pas の読み込みにより
NUnit 互換ファイルを出力

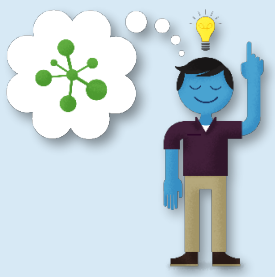
```
unit TestTCalcUnit;
interface
uses
    TestFramework, …, XMLTestRunner2;
type
    // テストメソッド
```

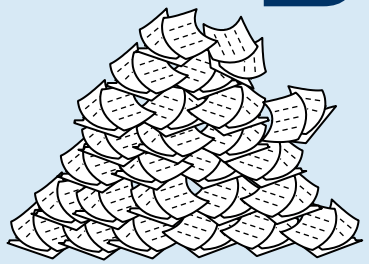
```
Initialization
    RegisterTest (TestTCalc.Suite);

With XMLTestRunner2.RunRegisteredTests do
Free;
end.
```

開発者とアプリ、要求とコードとビルドをつなぐ

Develop Build Deploy





開発者同士:

チーム同士:
開発者と、
- デザイナー
- テスター
- マネージャー

利害関係者:
開発チームと、
- 企画
- 運用
- 顧客

1 企画、アイデアの顕在化と実施計画

2 プロジェクト計画とタスクの割り出し

Bamboo
継続的インテグレーション
ビルドツール
ント
自動デプロイとデプロイ状況の管理

無償のデファクトスタンダード
なリポジトリ管理

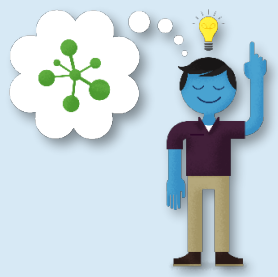
JIRA
BST / ITS: 要求、バグ、タスクの追跡, 変更管理
ドキュメントやソースコードの変更要素 (Issues) の追跡と管理
各種の成果物の粒度の調整と各成果物をつなぐ重要な役割
開発者とアプリの価値をわかりやすく示すのに欠かせない

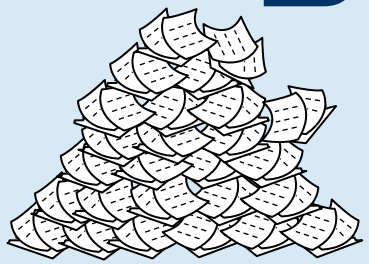
Confluence
ライブ ドキュメント共有
企画や仕様書を陳腐化させない。ドキュメントから協調、思考と経験の形式知化

HipChat
チーム内外のコミュニケーション インフラ
タイムラインでアクティビティを通知、対処の円滑化

開発者とアプリ、要求とコードとビルドをつなぐ

Develop Build Deploy





開発者同士:

チーム同士:
開発者と、
- デザイナー
- テスター
- マネージャー

利害関係者:
開発チームと、
- 企画
- 運用
- 顧客

SourceTree
DVCS クライアント
無償のデファクトスタンダード

Bamboo
継続的インテグレーション
ビルドツール
ント
自動ブロードキャスト ビルド状況の管理

3 開発作業の ”起動”

4 開発の実施

JIRA
各種の成果物の粒度の調整と各成果物をつなぐ重要な役割
開発者とアプリの価値をわか

5 継続的インテグレーション

Confluence
ライブ ドキュメント共有
企画や仕様書を陳腐化させない。ド

HipChat
チーム内外のコミュニケーション インフラ
タイムラインでアクティビティを通知、対処の円滑化

開発者とアプリ、要求とコードとビルドをつなぐ

Develop Build Deploy

開発者同士:

チーム同士:

開発者と、

- デザイナー
- テスター
- マネージャー

利害関係者:

開発チームと、

- 企画
- 運用
- 顧客

関連リソース

- Delphi と DUnit での継続的デリバリーについて
  http://re-workstyle.com/articles/continuous-integration-with-delphi-and-dunit/
- RAD Studio での Git 利用について
  http://re-workstyle.com/articles/rad-studio-git/
- DUnit の概要
  http://docwiki.embarcadero.com/RADStudio/XE5/ja/DUnit_%E3%81%AE%E6%A6%82%E8%A6%81
- DUnit での NUnit 互換結果ファイル出力 (XMLTestRunner2.pas)
  http://cc.embarcadero.com/Item/28239
- DUnitX
  http://www.finalbuilder.com/Resources/Blogs/PostId/697/introducing-dunitx

about Atlassian

We  Software

Atlassian Solution

ビジネス ／ 企画　　開発 ／ テスト

チーム ／ マネージメント　　運用 ／ ビジネス

ユニークなビジネスモデル

ビジネス モデル

- ✓営業 0 名
- ✓エキスパートによる付加価値
- ✓開発にフォーカス

成長し続ける

- ✓枯れた分野で急成長
- ✓ユーザーの支持が母体
- ✓イノベーションへの貢献

アトラシアンの文化を体験しに、遊びに来てください！

Google Maps: 「アトラシアン」

マリノスタウン内

最寄り駅: 各線 横浜駅から徒歩 10分

みなとみらい線 新高島から徒歩 5 分

アトラシアンの文化を体験しに、遊びに来てください！

アトラシアン製品

10 ユーザー $10 から

スターターライセンスは、全額をチャリティーに寄付

無償提供

OSS コミュニティや、クラスルームに無料で提供

全製品を無料で評価

30日フル機能を評価

オンプレミスとクラウド

ダウンロード版とオンデマンド版を提供

無料で講演いたします

出張セミナーなどお気軽にご連絡ください。

メール: tnagasawa@atlassian.com

Twitter: @tomohn

Facebook: Tomoharu.Nagasawa