SDLC

Engineering Life Cycle

Baby Cup で学ぶシステム開発

SDLCとは？

Software Development Life Cycle
ソフトウェアの「一生」のこと。

1. **設計** (Design)
2. **実装** (Implementation)
3. **運用** (Operation)
4. **改善** (Feedback)

> 💡 *Baby Cup のプロセスは、この **SDLC** の縮図です。*

Phase 1: 設計 (Design)

構築考察とチーム編成

構築考察 ≒ 要件定義

環境分析（メタ読み）

- 「今、何が流行っているか？」
- 市場調査なしで作るシステムは、誰にも使われない（勝てない）。

コンセプト決定

- 「どう勝つか」というコアバリューの定義。
- これがブレると、機能（技・努力値）が散漫になる。

チーム構成 ≒ アーキテクチャ

パーティの組み方は、システム構成そのもの。

ポケモン	エンジニアリング
役割分担	**責務の分離 (SoC)** 各コンポーネントが単一の役割を持つ
相性補完	**冗長化** 誰かが倒れても他でカバーする
1体への依存	**単一障害点 (SPOF)** そこが崩れると全システム停止

> ***Warning***: *「この1体が死んだら終わり」は **SPOF** です。*
> *良いパーティは、誰が倒れても立て直せる（**耐障害性が高い**）。*

Phase 2: 実装 (Implementation)

厳選と育成

厳選と努力値 ≒ 実装とチューニング

個体厳選 = コーディング

- 妥協した個体 ≒ **バグの潜むコード**
- 本番で予期せぬ挙動を引き起こす。

努力値振り = パフォーマンスチューニング

- 「S調整で先手を取る」
- 「レイテンシを削ってレスポンスを速くする」
- 全く同じ思考プロセス。

技術的負債の発生

> *「まあいいか」で妥協した個体や努力値調整は……*

📉 技術的負債 (Technical Debt)

大事な場面（昇格戦や大規模トラフィック時）で、
必ず足を引っ張ります。

- 1の差で競り負ける
- 想定外の攻撃を耐えられない

Phase 3: 運用 (Operation)

対戦本番

選出 ≒ リソース配分

6体 → 3体 の制約

限られたサーバリソースをどう配分するか？

- **選出読み**:
  相手（負荷状況）に合わせて、最適な3体（インスタンス）を選ぶ。
- **オートスケーリング**:
  状況に応じて必要なリソースを投入する能力。

立ち回り ≒ インシデント対応

対戦中（リリース後）は想定外のことばかり。

- 「読みが外れた！」 → **障害発生** 🚨
- 「急所に当たった！」 → **突発的スパイク** 📈

ここで問われるのは

- 瞬時の判断力
- パニックにならずにリカバリーする力 (**SRE力**)

交代 (Switching) ≒ 防御的運用

最強の防御行動：Workaround

不利な状況（バグ・相性不利）をまともに受けない。
「安全な場所へ逃げる（交代する）」 判断力。

1. **回避策 (Workaround)** を即座に打つ
2. 相手のリソース（交代権）を枯渇させる
3. 安全なタイミングで反撃する

> ***安定運用の鉄則です。***

Phase 4: 改善 (Feedback)

振り返りと次戦への準備

感想戦 ≒ ポストモーテム

負けた試合（障害）こそ、学びの宝庫。

Blameless（非難しない）な姿勢

重要なのは「誰が悪かったか」ではない。
「なぜ起きたか（仕組みの不備）」 を追求する。

- ❌「お前が交代しなかったから負けた」
- ⭕「交代のリスクを過大に見積もる**構築の欠陥**があったのでは？」

サイクルを回す

1. **Feedback**: 負けから学ぶ
2. **Design**: 構築を見直す
3. **Implementation**: 育成し直す
4. **Operation**: 次の対戦へ

🔄 アジャイル開発

この高速な改善サイクルこそが、
私たちが目指す **アジャイルな開発スタイル** そのものです。

まとめ

エンジニアリングとして楽しむ

1. **堅牢な設計** (Design)
    - SPOFをなくす
2. **妥協のない実装** (Implementation)
    - 技術的負債を残さない
3. **柔軟な運用** (Operation)
    - リスクを回避し続ける (Workaround)
4. **建設的な振り返り** (Feedback)
    - 失敗から学ぶ (Blameless)

> *ポケモンを通じて、私たちは*
> ***「システムを育て、守り、改善する力」*** *を養っています。*