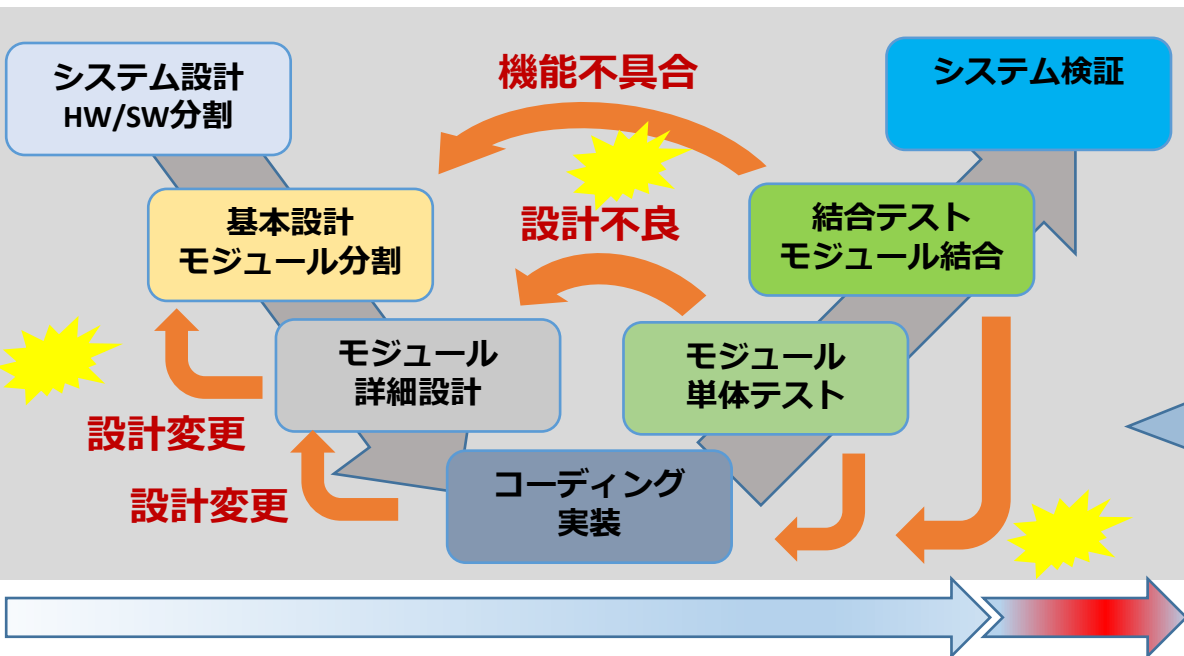
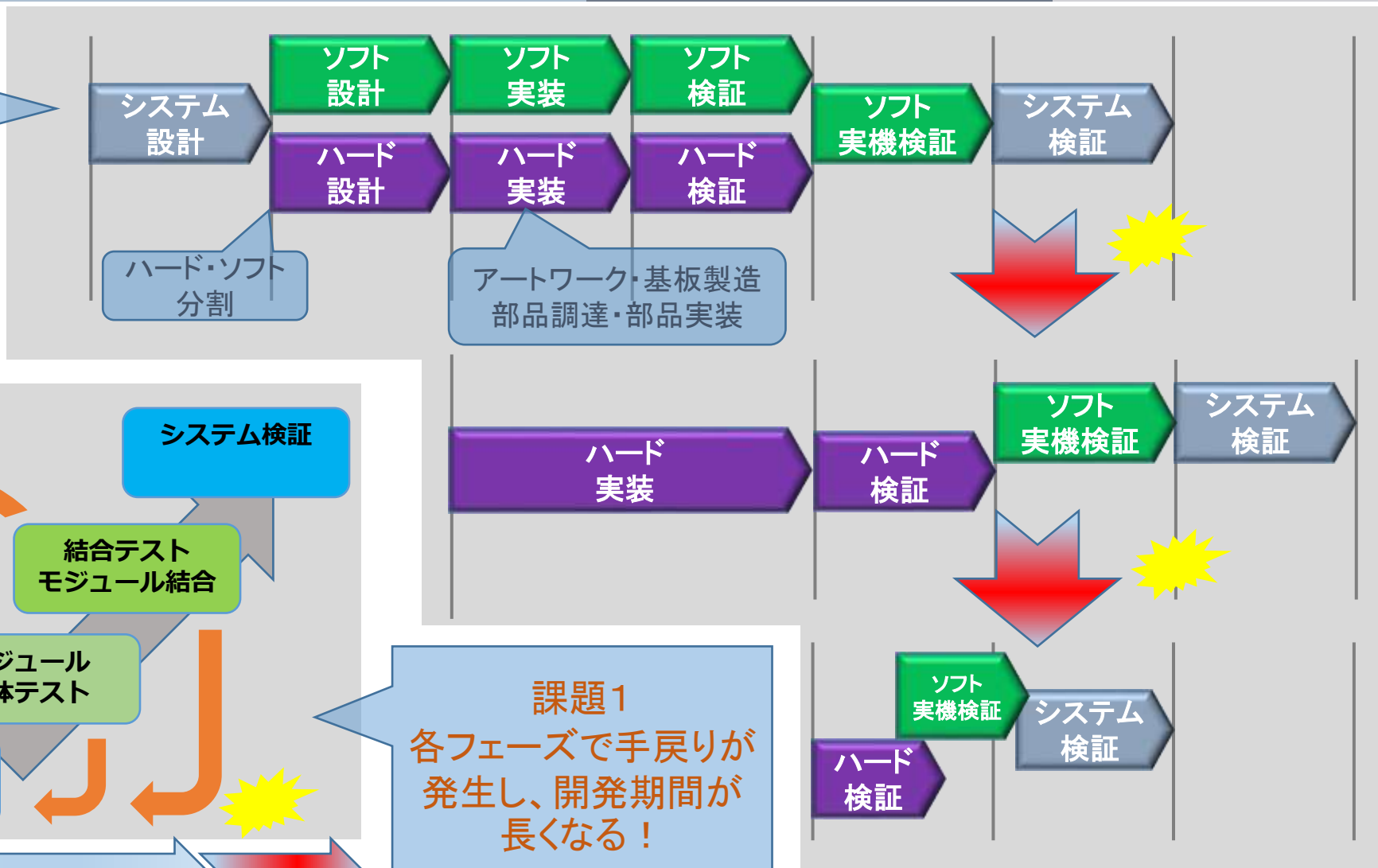


株式会社インターバディ

電子機器向け
組込ソフト開発の技、
あれこれ

電子機器 組込ソフト開発の課題

課題2
 往々にしてHW実装は遅れ、SW検証の開始も遅れる！
 リリース期日は変えられず、SW検証機能が削られる！



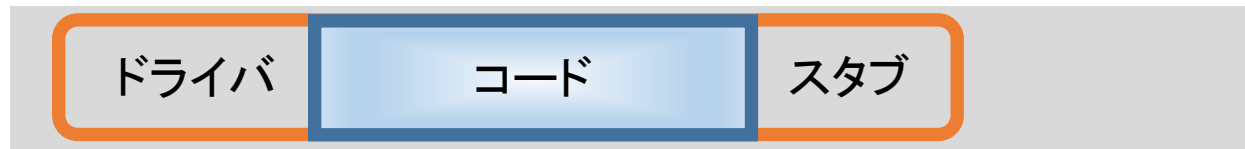
課題1：単体テストの徹底で解決！（技1）



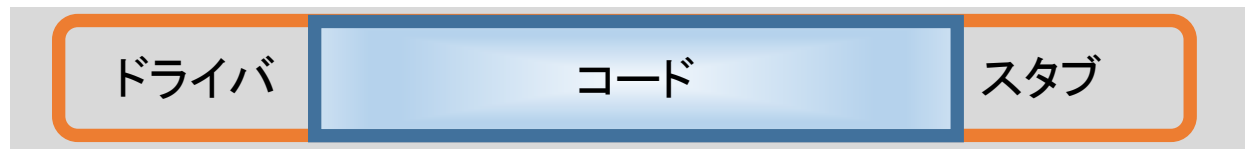
ドライバ：対象コードを呼び出す上位のモジュールのテスト用代替品
コード：実装する対象のプログラムソースコード
スタブ：対象コードから呼び出される下位モジュールのテスト用代替品



少ないコードで動かしてバグをつぶし、コードを書き足してはデバッグする。
エラー処理は後回しになり勝ちだが、早期にデバッグを完了した方が近道。



不完全なコードだと、再利用する際にバグを増やすことになる。
単体テストで、全てのコードを通すことも重要。



コードが完成しても、ドライバ、スタブは残しておく。
結合テストで不具合が出た場合も、単体テストに戻ってデバックする。

当たり前のことですが、単体テストで可能な限りバグを潰すことで、結合テスト、システム検証での不具合を少なくすることが出来ます。

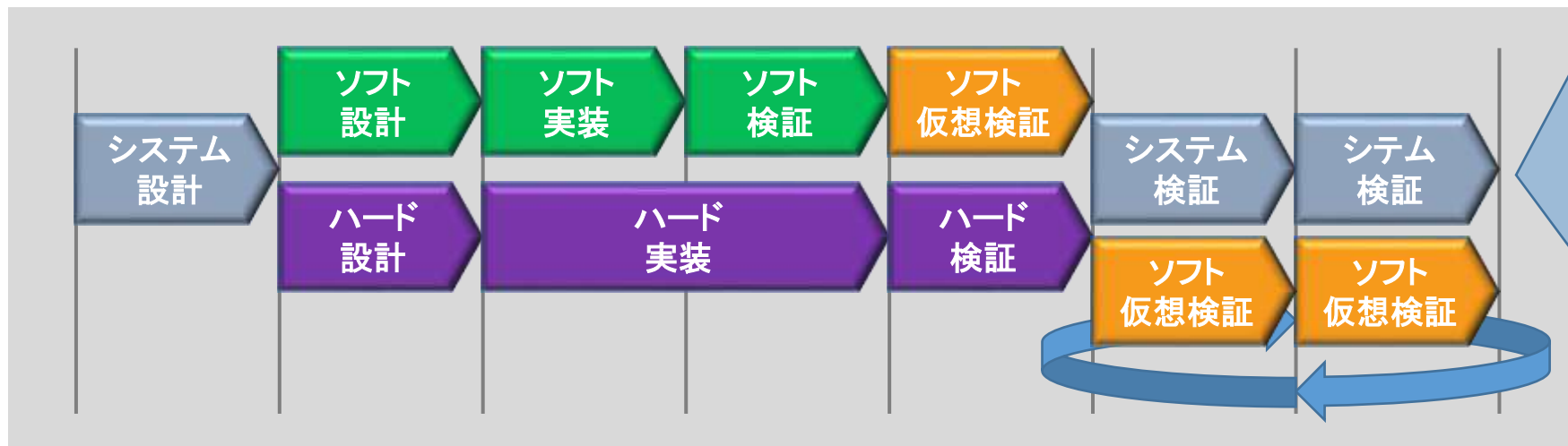
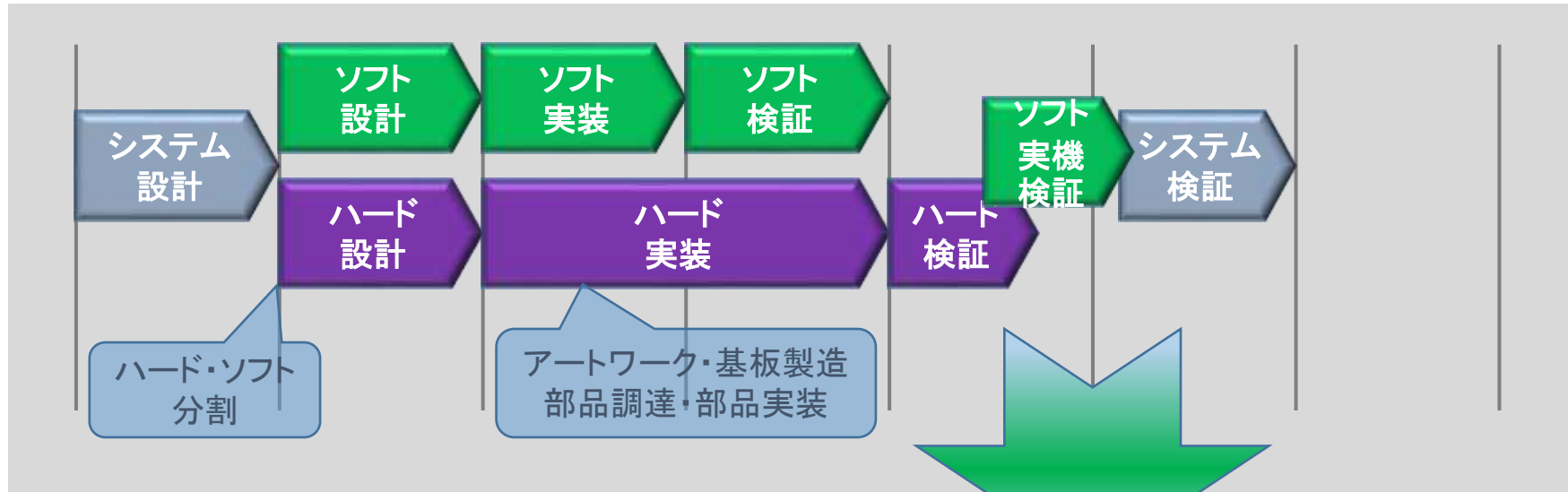
実装（コーディング）と単体テストは同時に行うこととなります。

→ スパイラル開発モデル

一見、ドライバとスタブを開発する工数が増えて、開発期間が伸びそうですが、結合テストやシステム検証を確実に完了するための近道となります。

結合テストでは、インターフェイスの確認に集中することができ、システム検証では、色々なユースケースでの検証に集中できます。

課題2：仮想検証で解決！（技2）



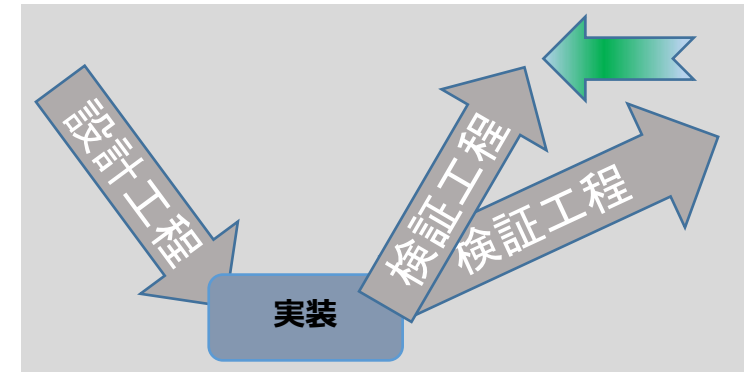
実機が入手できるまでに仮想環境で検証を開始！

初期開発のみならず、その後検証でも活躍！

- ◆”実機で動けばOK”
→ 波形で信号状態を視覚的に確認、科学的検証
 - ◆”仕様違反だが、実機ではたまたま動作”
→ 仕様違反を発見して警告
 - ◆レグレッションテスト
→ 多くのテストセットを仮想環境で自動化
→ 並列処理で検証TATも短縮
 - ◆実機では難しい故障注入
→ 仮想環境で容易に実現、再現性も確実
- 実機と仮想のバランスが重要

1) 徹底した単体テストで手戻りを最小限に
結合テスト、システム検証を容易にし、品質向上へ
#当たり前の話ですいません。

2) 仮想検証で、実機レス検証を開始
科学的検証、仕様確認、レグレッションテスト、故障注入でも威力を発揮
#ほんとはこっちをアピールしたかったんです。



#紙面の関係で多く掲載できなかったので...
詳しい情報は、下記URLをご参照下さい。

URL: <https://www.interbuddy.co.jp/jevec-day-2020-online>

展示Q&Aコーナー

時間になりましたら、インターバディ展示パネルの  をクリックして下さい。下記URLにつながります。

日時 : 2020年12月14日(月)16:20~17:00

URL : <https://zoom.us/j/91084923945?pwd=RzdMS3JtZ2VMbWJiWmR0dEd5QnlpUT09>

ミーティングID : 910 8492 3945

パスコード : jevec

Zoomミーティングにて、ご来訪をお待ち申し上げます。