

文書番号:RAC0112

開発工程表は誰のために書くか？

2006年2月

株式会社レンタコーチ

<http://homepage2.nifty.com/rent-a-coach/>

概要

- 対象者
 - ソフトウェア開発責任者
 - 又はその候補者
- 習得事項
 - 開発工程表を書く狙い
 - 開発工程表を書くためのコツ

目次

- 誰がなぜ書くか?
- 何を書くか?
- どう考えるか?
- どう表現するか?
- 進捗報告
- うまくいくためのヒント
- 演習問題

誰がなぜ書くか？

- 開発責任者が、開発を早く、間違いなく進めるため
- そのためには、
 - 最初に、開発手順を考える
 - 途中では、異常状態を早期発見
- 結果として、早く、問題なく完了

何を書くか?

- 構成要素
 - その商品を構成するものは何か?
 - どのようなモジュールに分解できる?
- 作業項目
 - やることは何?
 - 着手できる条件は何?
 - outputは何?どれくらいの量?
 - 誰が, 何時から始めて, 何時までに
 - リスクは何?
- 何時, どこへリリース?

どう考えるか？

- 最初は納期を気にせず，やるべきことをすべて洗い出す。
- やる内容，質，量を見積もる。
- 次に，納期をキープできるように，やる順序，やり方などを工夫する。
- 遅れのリスクを洗い出す。その回避策を用意する。

どう表現するか?

- 構成要素は階層的に表現し, 必要に応じて工程表も階層化する
- 作業項目はやる順序を明示
 - どの作業が終わらないと着手できないか
- 一番手間のかかるcritical pathを明示
- 作業項目の進捗の測り方
 - outputを定量的に見積もる
- 全体の進捗を測るcheck point
 - その通過条件と遅れたときの対策
- 進捗に影響を与えるリスク要因

		進度管理表(サンプル)						
構成要素と作業項目		担当	01/4	01/5	01/6	01/7	01/8	01/9
								某社へ出荷(8/6)
製品XY				DR-A(4/27)				
調査	甲		*	開発計画書50枚				DR-E(8/30)
組合せ試験	乙							80件
総合試験	乙							30件
試験環境構築	丙				***			DR-F(7/31)
ハード開発		(HW)				***		
モジュールA					DR-B(5/30)			
設計	イ		*	**	設計書100枚			
コーディング	イ							2KS
ハードと組合せ試験	イ					***		40件
								DR-D(6/29)
モジュールB								
設計	ロ		*	**	設計書300枚			
コーディング	ロ							10KS
単体試験	ロ							400件
マニュアル								
作成	ハ				**			200枚

進捗報告

- 工程表に進捗を記入するだけでは最良であるが、実際は、
- 開発状況表を併用
 - 工程表にある構成要素の進捗
 - 次週の予定、当面の目標の達成見通し
- 問題点管理表も併用
 - 問題発見時に、解決日限を決め、対策、要望などを記入
 - 経過状況を順に追記
 - 日限を越えたら、是正処置を講ずる

開発状況表

構成要素:XXXXXXXXXX		報告者:yyyyyy	
報告日	進捗状況	次週の予定	当面の目標と その見通し

問題点管理表

構成要素:XXXXXXXXXX			報告者:yyyyyy			
日付	問題点	対策または 要望	期限	経過	経過

うまくいくためのヒント

- Critical pathには精鋭を配置
- 遅れそうな前工程とはマージンを大きく
- 連携部門の遅れを監視し, 早期に対策
- 当初の開発工程表を変えるな
- Checkpoint未達成であれば, 対策を実施
- 進捗報告では, 工程表にないことは書かない, 書かせない

演習問題

- 新コンピュータ用にCコンパイラを開発
 - コンピュータAは新規開発
 - そのOSも新規開発(Cで書かれている)
 - コンピュータB用のCコンパイラのソースを購入し、それをAへ移植
- リスクは何か?

解答に代えて

- 構成要素
 - Parser , CG , runtime routine
 - Bで動くコンパイラ , Aで動くコンパイラ
- 作業項目
 - ソースコードの購入 , コンピュータBの購入
 - Bで動くA用のCGの作成
 - Bで動くコンパイラをAで動くように移植
 - Runtime routine作成

解答に代えて(その2)

- リスク
 - ソース購入契約の遅れ
 - コンピュータAの命令仕様の決定遅れ
 - Bで動くCGのインタフェース不整合
 - OS仕様の決定遅れ, デバグの長期化