

RUP 로 CMM 레벨 2, 3 도달

Rational Software 백서

TP 174

목차

요약.....	1
소개.....	1
레벨 2(반복 가능)	3
요구사항 관리.....	3
소프트웨어 프로젝트 계획	4
소프트웨어 프로젝트 추적 및 감시	5
소프트웨어 하청 관리	6
소프트웨어 품질 보증	6
소프트웨어 형상 관리(SCM)	7
레벨 3(정의됨).....	8
조직 프로세스 초점	8
조직 프로세스 정의.....	9
훈련 프로그램.....	9
통합 소프트웨어 관리	10
소프트웨어 제품 엔지니어링	10
그룹 간 조정.....	10
피어 검토.....	11
참조.....	11

요약

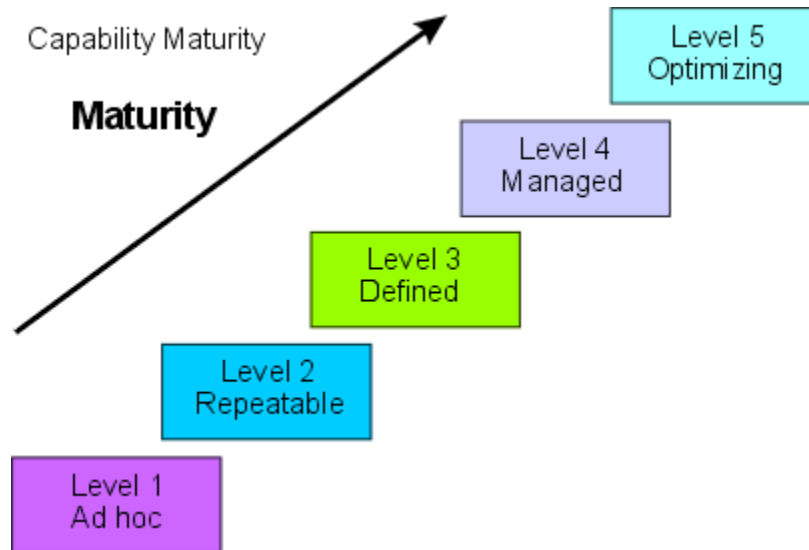
SEI(Software Engineering Institute)의 CMM(Capability Maturity Model)은 소프트웨어 프로세스 성숙도에 대해 가장 많이 참조되는 벤치마크 대상입니다. CMM 은 여러 분야에서 조직의 소프트웨어 프로세스 성숙도를 평가하기 위해 널리 사용되는 도구입니다. 이 백서는 CMM 레벨 2(반복 가능) 및 레벨 3(정의됨) 소프트웨어 프로세스 성숙도 레벨을 달성하려는 조직을 RUP 로 지원할 수 있는 방법에 대해 설명합니다.

소개

SEI(Software Engineering Institute)의 CMM(Capability Maturity Model)은 효과적인 소프트웨어 프로세스의 요소에 대해 설명하는 프레임워크입니다. 이 문서 끝 부분의 참조, [REF 1] 섹션을 참조하십시오. CMM 은 성숙되지 않은 임시 프로세스에서 성숙된 정규 프로세스로의 점진적인 개선 경로를 설명합니다.

CMM 은 소프트웨어 개발과 유지보수를 위한 계획, 엔지니어링 및 관리 사례에 대해 다룹니다. 이러한 핵심 사례는 비용, 스케줄, 기능성 및 제품 품질에 대한 목적을 달성할 수 있는 조직의 역량을 향상시킵니다.

CMM 의 성숙도 레벨은 레벨 1 에서 레벨 5 까지 5 단계입니다. 다음 그림과 같이 각 성숙도 레벨은 핵심 프로세스 영역(KPA)으로 구성되며 각 KPA 는 관련 활동의 클러스터를 식별합니다. 이러한 관련 활동을 함께 수행함으로써 해당 성숙도 레벨의 프로세스 기능을 달성하는 데 중요한 일련의 목적을 달성할 수 있습니다.



레벨 2(반복 가능 레벨)에서는 소프트웨어 프로젝트를 관리하는 정책과 해당 정책을 구현하는 프로시저가 설정됩니다. 새 프로젝트의 계획 및 관리는 유사 프로젝트에 대한 경험을 기반으로 합니다. 레벨 2 도달과 관련된 한 가지 목표는 소프트웨어 프로젝트에 대한 효과적인 관리 프로세스를 규정하는 것입니다. 조직은 이러한 프로세스를 통해 이전 프로젝트에서 개발된 성공 사례를 반복할 수 있습니다. 그러나 프로젝트에서 구현된 특정 프로세스는 다를 수 있습니다. 효과적인 프로세스는 실행됨, 문서화됨, 집행됨, 훈련됨, 측정됨 및 개선 가능 상태로 규정될 수 있습니다.

레벨 2 조직의 프로젝트에서는 기본 소프트웨어 관리 제어가 설치됩니다. 현실적인 프로젝트 약속은 이전 프로젝트에서 관찰한 결과와 현재 프로젝트의 요구사항을 기반으로 합니다. 프로젝트 추적 소프트웨어 비용, 스케줄 및 기능성과 약속 실천 문제점을 담당하는 소프트웨어 관리자는 실제로 발생할 때 식별됩니다. 소프트웨어 요구사항 및 이러한 요구사항을 충족시키기 위해 개발된 중간 산출물이 기준선이 되므로 해당

무결성은 제어됩니다. 소프트웨어 프로젝트 표준이 정의되며 조직에서 해당 표준을 성실하게 준수하는지 확인합니다. 소프트웨어 프로젝트는 해당 하청업체(있는 경우)와 협력하여 강력한 고객, 공급자 관계를 형성합니다.

레벨 2 조직의 경우 소프트웨어 프로젝트의 계획 및 추적 작업이 안정적이고 이전 성공을 반복할 수 있으므로 해당 소프트웨어 프로세스 기능을 규정된 기능으로 요약할 수 있습니다. 프로젝트 프로세스는 이전 프로젝트 성과를 기반으로 한 현실적인 계획에 따라 프로젝트 관리 시스템에서 효과적으로 제어합니다.

레벨 2의 KPA는 다음과 같습니다.:

- 요구사항 관리
- 소프트웨어 프로젝트 계획
- 소프트웨어 프로젝트 추적 및 감시
- 소프트웨어 하청 관리
- 소프트웨어 품질 보증
- 소프트웨어 형상 관리(SCM)

레벨 3(정의된 레벨)에서는 소프트웨어 엔지니어링 및 관리 프로세스를 모두 포함하여 조직에서 소프트웨어를 개발하고 유지보수하는 표준 프로세스가 문서화되며 해당 프로세스는 통합되어 전체적인 일관성을 나타냅니다. 표준 프로세스는 CMM 전체에서 조직의 표준 소프트웨어 프로세스로서 참조됩니다. 레벨 3에서 확정된 프로세스는 소프트웨어 관리자 및 기술 담당자의 보다 효과적인 역할 수행을 돕기 위해 사용(필요한 경우 변경)됩니다. 조직은 해당 소프트웨어 프로세스를 표준화할 때 실질적인 소프트웨어 엔지니어링 사례를 이용합니다. 조직의 소프트웨어 프로세스 활동(예: 소프트웨어 엔지니어링 또는 SEPG)을 담당하는 그룹이 있습니다. 담당자 및 관리자가 지정된 역할을 수행하는 데 필요한 지식과 스킬을 갖추도록 하기 위해 조직 전체를 대상으로 한 훈련 프로그램이 구현됩니다.

조직은 프로젝트를 통해, 정의된 자체 소프트웨어 프로세스를 개발하기 위한 표준 소프트웨어 프로세스를 사용자 조정하며 자체 소프트웨어 프로세스는 프로젝트의 고유 특성을 나타냅니다. CMM에서는 이렇게 사용자 조정된 프로세스를 프로젝트의 정의된 소프트웨어 프로세스라고 합니다. 정의된 소프트웨어 프로세스에는 잘 정의된 소프트웨어 엔지니어링 및 관리 프로세스의 일관된 통합 세트가 포함됩니다. 잘 정의된 프로세스는 작업 수행을 위한 준비 완료 기준, 입력, 표준 및 프로시저, 검증 메커니즘(예: 피어 검토), 출력 및 완료 기준을 포함하는 것으로 규정될 수 있습니다. 소프트웨어 프로세스는 잘 정의된 프로세스이므로 조직 경영진은 모든 프로젝트에 대한 기술적 진행상태를 정확하게 파악할 수 있습니다.

레벨 3 조직의 경우 소프트웨어 엔지니어링 및 관리 활동이 모두 안정적이고 반복 가능하므로 해당 소프트웨어 프로세스 기능을 일관적인 표준 기능으로 요약할 수 있습니다. 또한 구축된 제품 라인 범위에서 비용, 스케줄 및 기능성을 제어할 수 있으므로 소프트웨어 품질을 추적할 수 있습니다. 이 프로세스 기능은 정의된 소프트웨어 프로세스에서의 활동, 역할 및 책임에 대한 조직 전체의 공통 이해를 기반으로 합니다.

레벨 3의 KPA는 다음과 같습니다.

- 조직 프로세스 초점
- 조직 프로세스 정의
- 훈련 프로그램
- 통합 소프트웨어 관리
- 소프트웨어 제품 엔지니어링
- 그룹 간 조정
- 피어 검토

이 문서의 각 섹션은 RUP 기능, 방법, 프로시저 및 아티팩트가 KPA 목적을 충족시키는 방식에 대해 설명합니다.

이 문서는 CMM 프레임워크에서 레벨 2 및 레벨 3의 조직 성숙도 달성과 관련이 있는 조직 담당자를 대상으로 작성되었습니다.

레벨 2(반복 가능)

요구사항 관리

요구사항 관리의 목적은 고객과, 소프트웨어 프로젝트에서 처리하는 고객 요구사항에 대한 소프트웨어 프로젝트 사이에 공통 이해를 구축하는 것입니다. 고객과의 이러한 동의는 소프트웨어 프로젝트의 계획(소프트웨어 프로젝트 계획 KPA에서 설명) 및 관리(소프트웨어 프로젝트 추적 및 감시에서 설명)의 기반이 됩니다. 고객과의 관계 제어는 소프트웨어 형상 관리에서 설명하는 효과적인 변경 제어 프로세스 수행에 따라 결정됩니다.

RUP의 주요 특징 중 하나는 *유스 케이스 기반*이라는 점입니다. 유스 케이스는 사용자 요구사항의 도출, 구성 및 전달을 위한 조직적 접근 방식을 나타냅니다. 유스 케이스는 또한 프로젝트 개발, 테스트 및 반복 계획의 기반이 되는 기능적 요구사항을 문서화하는 방법을 제공합니다. RUP에서는 유스 케이스가 유스 케이스 모델에서 유지보수되며 분석, 테스트에서 유지보수까지 전체 프로젝트 라이프사이클에서 일관적으로 참조됩니다.

엔지니어링 관점에서 요구사항을 캡처하는 RUP 아티팩트는 다음과 같습니다.

- 유스 케이스 및 유스 케이스 패키지를 구성하는 유스 케이스 모델
- 비기능적 "보충 스펙"
- 유스 케이스 모델 조사
- 유스 케이스 보고서
- 용어집

개발될 유스 케이스 및 시나리오(요구사항)에 대해 설명하며 관리 측면에서 사용되는 RUP 아티팩트는 다음과 같습니다.

- 반복 계획
- 통합 빌드 계획
- 프로젝트 계획
- 소프트웨어 개발 계획

이러한 아티팩트는 모두 기준선이 되며 변경 관리 원칙을 따릅니다.

목적-1: 소프트웨어 엔지니어링 및 관리 사용을 위한 기준선을 설정하기 위해 소프트웨어에 할당되는 시스템 요구사항을 제어합니다.

RUP는 모든 전개 아티팩트의 진행 형상 제어를 지원하지만 '정규' 기준선은 다음 이정표와 일치합니다.

- 라이프사이클 목표 이정표(도입/인식(Inception) 단계)
- 라이프사이클 아키텍처 이정표(정제(Elaboration) 단계)
- 초기 운용 능력 이정표(구현/구축(Construction) 단계)
- 제품 릴리스 이정표(전이(Transition) 단계)

따라서 RUP는 요구사항, 해당 관리, 추적 및 기준선 작성에 대한 동의에 있어 CMM을 준수합니다.

목적-2: 소프트웨어에 할당된 시스템 요구사항과 소프트웨어 계획, 제품 및 활동을 일치시킵니다.

이 CMM 목적의 강조사항은 전달된 시스템이 사용자 요구사항을 충족시키도록 하는 것입니다. RUP 는 조직이 다음과 같은 두 가지 방식으로 이 목적을 달성할 수 있도록 지원합니다.

- *유스 케이스 접근 방식*은 사용자 요구 사항을 이해하고 캡처할 수 있도록 해줍니다. 캡처된 요구사항은 다양한 "비주얼" RUP 모델(유스 케이스, 디자인, 구현 및 테스트)로 전달되어 일관성을 유지하고 요구사항을 준수할 수 있습니다.
- *제어된 반복적 개발 접근 방식*은 프로젝트 위험성을 이해하고 조기에 발견한 후 자주 재검토하는 위험성 완화 접근 방식입니다. 진행되는 각 반복은 추가된 기능을 지속적으로 통합함으로써 위험성을 조기에 발견할 수 있습니다. 전통적인 폭포수형 방법론을 사용하는 경우 개발 라이프사이클의 후반부까지 이러한 위험성이 발견되지 않고 남아 있게 됩니다. 이처럼 위험성을 조기에 식별하는 경우 요구사항 범위를 재지정하거나 다른 기술적 변경사항을 적용함으로써 프로젝트 관리에 직접적인 이점을 제공할 수 있습니다.

RUP 관리 문서는 다음과 같습니다.

- 비즈니스 사례
- 소프트웨어 개발 계획
- 측정 계획
- 위험성 목록
- 프로젝트 계획
- 반복 계획
- 반복 평가 및 상태 평가

효과적인 변경 제어 및 관리는 지정, 추적 및 할당된 요구사항에 맞게 소프트웨어를 개발할 수 있는 또 다른 RUP 기능입니다.

RUP 는 각 프로젝트를 통해 개발 과정에서 발견되지 않은 결함 또는 제안된 변경사항의 범위와 영향(예산, 기술 또는 스케줄)을 조정할 수 있는 변경 제어 위원회(CCB)를 구성할 수 있도록 지원합니다. RUP 는 CCB 오퍼레이션을 지원하기 위해 강력한 형상 관리 및 버전 제어 도구/환경을 사용하도록 권장합니다.

소프트웨어 프로젝트 계획

소프트웨어 프로젝트 계획의 목적은 소프트웨어 엔지니어링 수행 및 소프트웨어 프로젝트 관리를 위한 올바른 계획을 수립하는 것입니다. 이 계획은 *소프트웨어 프로젝트 추적 및 감시* 섹션의 설명대로 소프트웨어 프로젝트를 관리하는 데 필요합니다. 효과적인 프로젝트 관리를 구현하려면 현실적인 계획을 수립해야 합니다.

목적-1: 소프트웨어 프로젝트의 계획 및 추적 시 사용할 소프트웨어 예산을 문서화합니다.

RUP 의 목적 중 하나는 모든 관계자의 기대가 동기화되어 일치하도록 하는 것입니다. 이 목적은 프로젝트 라이프사이클 전체에서 정기적인 평가를 수행함으로써 달성할 수 있으며 *상태 평가 보고서*에 문서화됩니다. 이 보고서는 자원(인력 구성 및 재무), 상위 10 가지 위험성, 메트릭을 통해 측정된 기술적 진행상태 및 주요 이정표 결과에 대한 데이터를 추적하기 위해 호출됩니다.

RUP 는 다음과 같은 메트릭 클래스를 이용합니다.

- 진행상태(코드 행, 클래스 수, 반복당 기능 점수, 재작업)
- 안정성(재작업 유형, 요구사항 또는 구현 휘발성)
- 적응성(재작업 비용)
- 모듈화(재작업 영향 정도)

- 품질(결함 발견 비율, 밀도, 상숙 깊이, 재작업 지표)
- 성숙도(실패당 테스트 시간)
- 자원에 대한 비용 프로파일(계획 대비 실제)

목적-2: 소프트웨어 프로젝트 활동 및 약속 계획을 수립하고 문서화합니다.

프로젝트 계획 및 약속을 캡처하는 RUP 문서는 다음과 같습니다.

- 비즈니스 사례
- 소프트웨어 개발 계획
- 측정 계획
- 위험성 목록
- 프로젝트 계획
- 반복 계획
- 반복 평가
- 상태 평가

목적-3: 관련 그룹 및 개인이 소프트웨어 프로젝트와 관련된 자신의 약속에 동의합니다.

RUP 에서 소프트웨어 개발 계획은 프로젝트에 대한 전반적인 계획을 정의하며 반복 계획은 반복에서 수행될 내용을 세부적으로 정의합니다. RUP 에서 필요한 반복 계획 검토는 반복이 시작되기 전에 합의가 이루어질 수 있도록 반복 계획을 모든 이해 당사자(stakeholder)에게 공개합니다. 프로젝트 관리자는 합의된 반복 계획에 따라 일련의 작업 지시를 생성합니다. 작업 지시는 관련 프로젝트 팀 및 개인에게 반복 계획의 의도를 세부적으로 알려줍니다. 프로젝트 관리자는 해당 작업 지시에 대해 관련 인력과의 동의를 이끌어냄으로써 반복을 진행합니다.

소프트웨어 프로젝트 추적 및 감시

소프트웨어 프로젝트 추적 및 감시의 목적은 소프트웨어 프로젝트의 성과가 소프트웨어 계획과 현저히 다를 때 경영진이 적절한 조치를 수행할 수 있도록 실제 진행상태를 정확하게 나타내는 것입니다.

목적-1: 소프트웨어 계획에 대한 실제 결과 및 성과를 추적합니다.

소프트웨어 프로젝트 계획 섹션의 설명대로, RUP 에는 여러 가지 프로젝트 계획 레벨과, 계획 대비 실제 성과를 평가하기 위한 *상태 평가 보고서*가 포함됩니다. 이 보고서는 특정 이정표에 대해 생성되며 프로젝트 관리자가 담당합니다.

주요 RUP 이정표는 단계(도입/인식(Inception), 정제(Elaboration), 구현/구축(Construction) 또는 전이(Transition)) 종료 시점과 일치하며 완료 기준이 잘 지정되어 있습니다. 검토 기회는 특정 단계에서 각 반복 종료 시점의 보조 이정표에 존재하며 향후 방향 제시를 위해 얻은 교훈 및 결정 지점으로 활용할 수 있습니다.

예를 들어, 정제(Elaboration) 단계의 목적은 문제점 도메인을 분석하고 올바른 아키텍처 기반을 마련하며 프로젝트 계획을 개발하고 프로젝트에서 가장 위험성이 큰 요소를 제거하는 것입니다. 아키텍처 관련 결정은 전체 시스템에 대한 이해를 기반으로 이루어져야 합니다. 즉, 특정 제한조건(보충 요구사항)을 고려하여 대부분의 유스 케이스에 대해 설명해야 합니다. 아키텍처를 확인하기 위해 아키텍처 관련 선택사항을 예시하고 중요한 유스 케이스를 실행하는 시스템이 구현됩니다.

정제(Elaboration) 단계 종료 시점에서는 세부 시스템 목표 및 범위와 함께, 아키텍처 선택사항 및 주요 위험성에 대한 해결책을 점검합니다. 실제 결과 및 성과가 소프트웨어 계획과 현저히 다른 경우 완료를 위한 정정 조치를 수행하고 관리해야 합니다.

*위험성 목록*은 프로젝트에 대해 알려진 모든 위험성의 개요 정보를 제공하는 RUP 아티팩트로서, 계획 및 프로젝트 평가를 위한 입력으로 이용됩니다. 각 위험성은 해당 영향과, 해당 위험성을 완화시키기 위해 호출되는 비상사태 계획의 관점에서 설명됩니다. 위험성 목록은 비즈니스 사례와 함께 개발되어 프로젝트에 대한 "진행" 또는 "중단" 결정의 기반이 됩니다. 위험성 목록은 프로젝트 라이프사이클 전체에서 유지보수됩니다.

목적-2: 실제 결과 및 성과가 소프트웨어 계획과 현저히 다른 경우 완료를 위한 정정 조치를 수행하고 관리해야 합니다.

RUP에서는 두 가지 레벨의 성능 추적이 필요합니다. 즉, 프로젝트 관리자는 프로젝트의 루틴 모니터링 및 제어의 일부로서, 수집된 메트릭을 사용하여 상태 평가를 생성합니다. 상태 평가에서 식별된 문제점은 프로젝트 관리자 작업 지시를 통해 직접 프로젝트에 대해 또는 변경 요청을 통해 보다 광범위한 문제점에 대해 문제점 해결 계획에 따라 처리됩니다. 또한 각 반복의 결과는 반복 평가 및 검토의 대상이 됩니다. 이를 통해 이전 반복 경험으로 후속 반복에 대한 계획을 안내하고 변경 요청을 통해 정정 조치를 관리할 수 있습니다.

목적-3: 관련 그룹 및 개인으로부터 소프트웨어 약속 변경사항에 대한 동의를 얻습니다.

이해 당사자(stakeholder)는 RUP의 설명에 따라 제어된 반복적 개발 프로세스를 통해 올바른 프로젝트 진행에 필요한 모든 변경사항 및 프로젝트 진행상태를 정기적으로 확인할 수 있습니다. 변경 제어 위원회(CCB)는 제안된 변경사항을 검토하여 전체 프로젝트 스케줄에 적용할 수 있는 현실적인 변경사항인지 확인합니다.

소프트웨어 하청 관리

하청 관리의 목적은 자격이 있는 소프트웨어 하청업체를 선택하여 효과적으로 관리하는 것입니다. 하청 관리에서는 기본 관리 제어에 필요한 요구사항 관리, 소프트웨어 프로젝트 계획과 소프트웨어 프로젝트 추적 및 감시의 관심사항을 소프트웨어 품질 보증 및 소프트웨어 형상 관리를 위해 필요한 조정과 결합하고 이 제어를 필요에 따라 하청업체에 적용합니다.

목적-1: 주 계약자가 자격이 있는 소프트웨어 하청업체를 선택합니다.

목적-2: 주 계약자와 소프트웨어 하청업체가 상호 약속에 동의합니다.

목적-3: 주 계약자와 소프트웨어 하청업체가 지속적인 커뮤니케이션을 유지합니다.

목적-4: 주 계약자가 소프트웨어 하청업체의 실제 결과와 성과를 해당 약속과 비교하여 추적합니다.

이러한 목적은 RUP의 현재 범위에 포함되지 않는 것으로서 조직에 따라 다릅니다.

하청 계약에 대해서는 RUP에서 구체적으로 다루지 않지만 해당 도구, 기법 및 메커니즘이 하청업체에도 적용되어 동일한 프로세스 상태를 유지하는 것으로 가정합니다.

모든 하청업체 결정사항은 비즈니스 사례에 문서화되어야 합니다. 주 계약자와 동일한 개발 계획을 수행하는 하청업체는 또한 동일한 기술 교환, 주요 이정표 및 상태 평가에 참여합니다.

소프트웨어 품질 보증

소프트웨어 품질 보증의 목적은 경영진이 소프트웨어 프로젝트에서 사용하는 프로세스와 빌드 대상 제품을 정확하게 파악할 수 있도록 하는 것입니다. 소프트웨어 품질 보증은 대부분의 소프트웨어 엔지니어링 및 관리 프로세스의 핵심 부분입니다.

RUP에서는 품질이 모든 프로젝트 인력의 공동 책임이며 어떤 조직에도 본질적으로 포함되지 않는 것으로 간주합니다.

목적-1: 소프트웨어 품질 보증 활동을 계획합니다.

소프트웨어 품질 보증 계획 타스크는 조직의 책임입니다. 그러나 RUP 에는 효과적인 프로젝트 품질 보증 프로그램을 구성할 수 있는 여러 가지 속성이 있습니다.

각 RUP 이정표에는 감사의 기초로 활용할 수 있는 특정 완료 기준이 있습니다. 또한 RUP 에는 별도 검토 타스크가 있습니다. 각 검토에는 다음 활동을 시작하기 위해 "패스"해야 하는 "게이트"를 나타내는 일련의 체크포인트가 연관됩니다.

RUP 는 특정 아티팩트를 검토해야 하는 담당자에 대한 안내 정보를 제공합니다. 예를 들어, 디자이너가 수행하는 "유스 케이스 분석"의 결과는 독립적인 설계자, 디자이너, 유스 케이스 디자이너 및 디자인 검토자가 검토해야 합니다. 정의된 RUP 및 아티팩트 검토 기준을 고려하여, 제품 품질과 관련된 목표로 프로세스 준수와 개발 표준 및 가이드라인에 대한 준수 여부를 쉽게 평가할 수 있어야 합니다.

목적-2: 적용되는 표준, 프로시저 및 요구사항에 대한 소프트웨어 제품 및 활동의 준수 여부를 객관적으로 확인합니다.

이 목적은 조직의 품질 담당자를 채용함으로써 달성할 수 있습니다. 그러나 RUP 는 프로젝트 표준으로 적용할 수 있는 필수 검토 체크리스트와 문서 템플릿을 제공합니다.

목적-3: 관련 그룹 및 개인에게 소프트웨어 품질 보증 활동과 해당 결과를 알려줍니다.

소프트웨어 프로젝트 계획의 설명대로, RUP 목적 중 **하나**는 모든 관계자의 기대가 동기화되어 일치하도록 하는 것입니다. RUP 는 품질 감사 결과의 입력과 별도로 자원(인력 구성 및 재무), 상위 10 가지 위험성, 메트릭을 통해 측정된 기술적 진행상태 및 주요 이정표 결과에 대한 보고 시 호출됩니다. RUP 메트릭 프로그램은 다음 메트릭 수집에 대한 가이드라인을 제공합니다.

- 진행상태(코드 행, 클래스, 반복당 기능 점수)
- 안정성(재작업 유형, 휘발성)
- 적응성(재작업 비용)
- 모듈화(재작업 영향 정도)
- 품질(결함 발견 비율, 밀도, 상속 깊이)
- 성숙도(실패당 테스트 시간)
- 비용 프로파일(계획 대비 실제)

목적-4: 소프트웨어 프로젝트에서 해결할 수 없는 비준수 문제는 상위 경영진에서 처리합니다.

이 내용은 RUP 의 범위에 포함되지 않으며 조직의 책임입니다. 그러나 RUP 에서 설명하는 변경 제어 프로세스에서는 비준수 문제를 해결하기 위해 문서화하고 확대하는 메커니즘을 사용할 수 있습니다.

소프트웨어 형상 관리(SCM)

소프트웨어 형상 관리의 목적은 프로젝트의 소프트웨어 라이프사이클 전체에서 소프트웨어 프로젝트의 제품 무결성을 확립하고 유지보수하는 것입니다. 소프트웨어 형상 관리의 대부분의 소프트웨어 엔지니어링 및 관리 프로세스의 핵심 파트입니다.

목적-1: 소프트웨어 형상 관리 활동을 계획합니다.

RUP 의 설명대로, 올바른 형상 관리는 제어된 반복적 개발 방법에 있어 필수 요소입니다. 소프트웨어는 단계적으로 전개되므로 후속 개발을 위해서는 이전 개발 노력의 소프트웨어 버전을 사용할 수 있어야 합니다. 각 단계에서 작업 소프트웨어를 명확하게 생성하는 방법을 계획하는 것은 RUP 의 핵심 작업입니다.

RUP에는 프로젝트의 소프트웨어 개발 자원을 유지보수하는 방법과 해당 자산을 통합하는 방법을 정의하는 다음과 같은 두 가지 기본 도구가 있습니다.

- 형상 관리 계획
- 통합 빌드 계획

형상 관리 계획은 도입/인식(Inception) 단계에서 시작되며 다음 내용을 설명합니다.

- 소프트웨어 버전화 관리 및 처리
- 해당 RUP 모델 저장 및 형상 항목으로 나누기
- 변경 제어 방법을 사용하여 변경사항 및 릴리스 관리

통합 빌드 계획은 빌드될 형상 항목에 대한 세부사항과 형상 항목을 특정 반복에 통합하는 순서를 제공합니다.

목적-2: 선택한 소프트웨어 중간 산출물을 식별, 제어하고 사용 가능하게 합니다.

RUP 형상 관리 계획은 형상 제어 및 관리 프로세스에 대한 설명을 요구하여 중간 산출물을 실제로 식별, 제어 및 사용할 수 있도록 합니다.

목적-3: 식별된 소프트웨어 중간 산출물에 대한 변경사항을 제어합니다.

RUP는 프로젝트에서 변경 제어 위원회(CCB)를 유지보수하고 변경 관리 시스템을 통해 변경 요청사항을 올바르게 관리, 추적 및 구현하고 해당 비용을 산정할 수 있도록 지원합니다.

목적-4: 관련 그룹 및 개인에게 소프트웨어 기준선의 상태와 콘텐츠를 알려줍니다.

RUP는 요구사항, 디자인 및 구현 기준선과 이들 간의 추적성을 전자 형식으로 유지할 수 있도록 지원합니다. 다양한 레벨의 프로젝트 제어로 기준선 변경사항을 조정합니다. 예를 들어, 변경 제어 위원회(CCB)는 요구사항 레벨 변경사항의 영향을 고려합니다. 범위가 넓지 않은 디자인 및 구현 변경사항은 기술 기관의 해당 레벨에서 검토합니다. 승인, 제어 레벨 및 이들을 전달하는 방식은 *형상 관리 계획*과 *소프트웨어 개발 계획*에서 설명합니다.

레벨 3(정의됨)

조직 프로세스 초점

조직 프로세스 초점의 목적은 조직의 전체 소프트웨어 프로세스 역량을 개선하는 소프트웨어 프로세스 활동에 대한 조직 책임을 설정하는 것입니다. 조직 프로세스 초점의 기본 결과는 일련의 소프트웨어 프로세스 자산이며 이 내용은 조직 프로세스 정의에서 설명합니다. 이러한 자산은 통합 소프트웨어 관리의 설명대로 소프트웨어 프로젝트에서 사용합니다.

목적-1: 소프트웨어 프로세스 개발 및 개선 활동은 조직 전체에서 조정됩니다.

RUP는 정의된 동일한 프로세스를 여러 반복에서 재규정하는 반복 프로세스입니다. 프로세스 규정의 이러한 반복적 특성과 각 단계 및 반복에서 얻은 교훈과 상태 메트릭의 평가를 통해 각 연속 반복 프로세스를 세부적으로 조정할 수 있는 기회를 제공합니다.

목적-2: 소프트웨어 프로세스의 강점과 약점은 프로세스 표준에 따라 식별됩니다.

RUP는 특정 프로젝트 유형에 효과적으로 사용하도록 사용자 조정할 수 있는 전체 소프트웨어 개발 프로세스를 나타냅니다. RUP 사용자 조정 방법에 대한 안내 정보는 *환경 원칙*에서 제공합니다. 기술 및 관리 복잡도 이외에, 프로젝트에서 사용할 프로세스의 "모양"을 결정할 일부 프로세스 판별은 다음과 같습니다.

- 비즈니스 컨텍스트(계약, 추론 또는 내부)
- 소프트웨어 개발 노력의 규모
- 혁신 정도
- 응용프로그램 유형

목적-3: 조직 레벨의 프로세스 개발 및 개선 활동을 계획합니다.

이 레벨 3 목적은 전적으로 채택 조직에 의존합니다.

조직 프로세스 정의

조직 프로세스 정의의 목적은 프로젝트 간 프로세스 성능을 개선하고 조직에 지속적으로 장기적인 이익의 기반을 제공하는 재사용가능한 소프트웨어 프로세스 자산 세트를 개발 및 유지보수하는 것입니다. 이러한 자산은 훈련(훈련 프로그램에서 설명)과 같은 메커니즘을 통해 규정할 수 있는 안정적인 기반을 제공합니다.

목적-1: 조직의 표준 소프트웨어 프로세스를 개발하고 유지보수합니다.

RUP 는 순조로운 출발을 제공할 수 있으며 전개, 사용자 조정 및 유지보수할 수 있는 조직의 기준선 소프트웨어 개발 프로세스가 됩니다.

목적-2: 소프트웨어 프로젝트에서 조직의 표준 소프트웨어 프로세스 사용과 관련된 정보를 수집, 검토하고 사용 가능하게 만듭니다.

이 목적은 RUP 를 채택하는 조직에서 지원해야 합니다.

훈련 프로그램

훈련 프로그램의 목적은 개인이 해당 역할을 효과적이고 효율적으로 수행하는 데 필요한 스킬과 지식을 개발하는 것입니다. 훈련은 조직의 책임이지만 소프트웨어 프로젝트에서 필요한 스킬을 식별해야 하며 프로젝트 요구가 고유한 경우 필요한 훈련을 제공해야 합니다.

목적-1: 훈련 활동을 계획합니다.

이 목적은 RUP 를 채택하는 조직에서만 달성할 수 있습니다. 그러나 RUP 는 다양한 소프트웨어 개발 활동을 수행하는 방법에 대한 가이드라인, 개념 및 단계별 세부 설명을 제공하는 "업계 우수 사례"에 대한 지식 기반입니다. 따라서 RUP 는 그 자체만으로도 훌륭한 훈련 자료 소스가 됩니다.

RUP 관련 지원 과정은 다음과 같습니다.

- 요구사항, 분석 및 디자인, 구현, 테스트, 아키텍처, 프로세스 구성, 관리, 도구 및 객체 지향 소개에 대한 모듈을 통한 RUP 개요
- 유스 케이스를 통한 요구사항 관리(RMUC)
- 객체 지향 프로젝트 관리(OOPM)
- 객체 지향 디자인 및 분석(OOAD)
- 자동화 소프트웨어 품질
- 형상 관리
- 소프트웨어 아키텍처 및 반복 프로세스

목적-2: 소프트웨어 관리 및 기술 역할을 수행하는 데 필요한 스킬과 지식을 개발하기 위한 훈련을 제공합니다.

목적-3: 소프트웨어 엔지니어링 그룹 및 소프트웨어 관련 그룹에 속하는 개인은 해당 역할을 수행하는 데 필요한 훈련을 받습니다.

이러한 훈련 프로그램의 목적은 RUP 를 채택하는 조직에서 달성해야 합니다. 그러나 RUP 는 이전 섹션에서 설명한 대로 일련의 과정을 제공합니다.

통합 소프트웨어 관리

통합 소프트웨어 관리의 목적은 소프트웨어 엔지니어링 및 관리 활동을 조직의 표준 소프트웨어 프로세스 및 관련 프로세스 자산에서 사용자 조정되는 일관되고 정의된 소프트웨어 프로세스로 통합하는 것입니다. 프로세스 자산에 대한 설명은 *조직 프로세스 정의* 섹션에서 제공합니다. 이 사용자 조정은 소프트웨어 제품 엔지니어링의 설명대로 프로젝트의 비즈니스 환경 및 기술 요구를 기반으로 합니다. 통합 소프트웨어 관리는 소프트웨어 프로젝트 계획에서 레벨 2 의 소프트웨어 프로젝트 추적 및 감시로 전개됩니다.

목적-1: 프로젝트의 정의된 소프트웨어 프로세스는 조직의 표준 소프트웨어 프로세스의 사용자 조정 버전입니다.

RUP 의 표준 전달은 RUP *환경 원칙에 따라* 구성할 수 있으며 다양한 유형의 프로젝트에서 사용할 수 있도록 범위를 재지정할 수 있습니다.

목적-2: 프로젝트의 정의된 소프트웨어 프로세스에 따라 프로젝트를 계획하고 관리합니다.

이 목적은 RUP 를 채택하는 조직에서 처리해야 합니다.

소프트웨어 제품 엔지니어링

소프트웨어 제품 엔지니어링의 목적은 모든 소프트웨어 엔지니어링 활동을 통합하는 잘 정의된 엔지니어링 프로세스를 지속적으로 수행함으로써 정확하고 일관된 소프트웨어 제품을 효과적이고 효율적으로 생성하는 것입니다. 소프트웨어 제품 엔지니어링은 프로젝트의 기술 활동(예: 요구사항 분석, 디자인, 코드 및 테스트)에 대해 설명합니다.

목적-1: 소프트웨어 엔지니어링 타스크를 정의, 통합하고 지속적으로 수행함으로써 소프트웨어를 생성합니다.

각 역할에서 필요한 작업에 대한 RUP 타스크 및 정의는 필수 프로젝트 계획 아티팩트의 배경에 따라 타스크가 정의, 할당 및 완료될 수 있도록 합니다. RUP 의 고유한 반복적 개발 프로세스는 소프트웨어 개발 팀의 효율성을 빠르게 입증하고 최종 제품에 대한 평가를 제공합니다.

목적-2: 소프트웨어 제품을 서로 일치되도록 유지합니다.

환경에서 엔지니어링 모델(예: 유스 케이스 모델, 디자인 모델, 소스 코드 및 실행 가능 컴포넌트) 간의 추적성을 유지보수합니다.

그룹 간 조정

그룹 간 조정의 목적은 소프트웨어 엔지니어링 그룹이 다른 엔지니어링 그룹과 적극적인 관계를 유지할 수 있는 방법을 확립함으로써 프로젝트가 고객의 요구를 보다 효율적이고 효과적으로 충족시킬 수 있도록 하는 것입니다. 그룹 간 조정은 소프트웨어 엔지니어링의 범위를 벗어난 통합 소프트웨어 관리의 제휴적 측면입니다. 따라서 소프트웨어 프로세스를 통합하고 다른 그룹과 소프트웨어 엔지니어링 그룹의 상호 작용을 조정하고 제어해야 합니다.

목적-1: 모든 관련 그룹이 고객의 요구사항에 동의합니다.

다른 "정규" 요구사항 스펙 방법에 비해, 유스 케이스를 요구사항 캡처 및 설명의 기반으로 사용하는 데 따른 실질적인 이점 중 하나는 관련 이해 당사자(stakeholder)가 유스 케이스를 쉽게 이해할 수 있다는 것입니다.

따라서 유스 케이스 요구사항을 캡처하는 RUP 방법은 모든 이해 당사자가 수행되어야 할 작업에 대해 동의할 수 있음을 의미합니다. 이 방법은 프로세스에서 계속 수행되며 소프트웨어 개발을 위한 기반으로 사용되는 모델 및 검토에 반영됩니다.

목적-2: 관련 그룹이 엔지니어링 그룹 간의 약속에 동의합니다.

이 목적은 RUP 를 채택하는 조직에서 처리해야 합니다. 그러나 RUP 비주얼 모델을 통해 요구사항 캡처에서 배치까지 각 제품 개발 단계에서 필요한 내용을 쉽게 이해할 수 있습니다. 또한 RUP 변경 및 형상 관리 프로세스를 통해 제안된 변경사항을 올바르게 평가하고 모든 이해 당사자(stakeholder)에게 전달할 수 있습니다. 엔지니어링 그룹은 그룹 간 문제를 식별, 추적 및 해결합니다. RUP 의 반복적 개발 프로세스에서는 개발된 모든 소프트웨어에 대한 통합 진행을 통해 소프트웨어 문제점을 조기에 발견할 수 있습니다. 여러 팀에서 개발한 소프트웨어의 통합 문제점은 팀 간 문제를 발생시키고 해결하기 위한 "공통 영역"의 기능을 수행할 수 있습니다. 이 개념은 프로젝트 개발 문제를 캡처, 추적 및 해결하기 위한 정규 메커니즘을 제공하는 RUP 결함 및 변경 요청 프로세스에서 지원합니다.

피어 검토

피어 검토의 목적은 소프트웨어 중간 산출물의 결함을 조기에 효율적으로 제거하는 것이며 결과적으로 소프트웨어 중간 산출물 및 예방할 수 있는 결함을 보다 정확하게 이해할 수 있습니다. 피어 검토는 소프트웨어 제품 엔지니어링에서 호출되는 효과적이면서도 중요한 엔지니어링 방법입니다.

목적-1: 피어 검토 활동을 계획합니다.

레벨 2 의 품질 보증 목적에서 설명한 대로 RUP 내 각 TASK에는 개별 검토 TASK가 포함됩니다.

문제점을 조기에 발견함으로써 전체 비용을 줄일 수 있으므로 RUP 는 모든 아티팩트, 특히 중요한 아티팩트에 대한 "많은 초기" 피어 검토를 지원합니다. RUP 는 각 단계 및 각 모델에서 검토할 특징적인 기능에 대한 체크리스트를 제공합니다.

목적-2: 소프트웨어 중간 산출물의 결함을 식별하고 제거합니다.

RUP 아티팩트 검토자는 해당 아티팩트를 다음 개발 단계에 적용할 수 있는지 여부를 판별해야 합니다. 아티팩트가 검토 "패스" 기준을 충족시키지 못하는 경우 RUP 메트릭 프로그램에 따라 다음 특성에 대한 세부사항을 캡처해야 합니다.

- 안정성(재작업 유형, 휘발성)
- 적응성(재작업 비용)
- 모듈화(재작업 영향 정도)
- 품질(결함 발견 비율, 밀도, 상속 깊이)
- 성숙도(실패당 테스트 시간)
- 비용 프로파일(계획 대비 실제)

참조

[REF1] Mark C. Paulk 외, "Key Practices of the Capability Maturity Model – Version 1.1", Software Engineering Institute—Carnegie Mellon University



본사 안내:

Rational Software
18880 Homestead Road
Cupertino, CA 95014
전화번호: (408) 863-9900

Rational Software
20 Maguire Road
Lexington, MA 02421
전화번호: (781) 676-2400

수신자 부담 전화번호: (800) 728-1212

전자 우편: info@rational.com

웹: www.rational.com

전 세계 지사 안내: www.rational.com/worldwide

Rational, Rational 로고 및 Rational Unified Process 는 미국 또는 기타 국가에서 사용되는 Rational Software Corporation 의 등록상표입니다. Microsoft, Microsoft Windows, Microsoft Visual Studio, Microsoft Word,

Microsoft Project, Visual C++ 및 Visual Basic 은 Microsoft Corporation 의 상표 또는 등록상표입니다. 기타 다른 이름들은 식별용으로만 사용되며 해당 회사의 상표 또는 등록상표입니다. ALL RIGHTS RESERVED. Made in the U.S.A.

© Copyright 2002 Rational Software Corporation.

본 내용은 통지 없이 변경될 수 있습니다.