

Fachhochschule Frankfurt am Main
Fachbereich 2: Informatik
SS 2008

IT Projekt Management

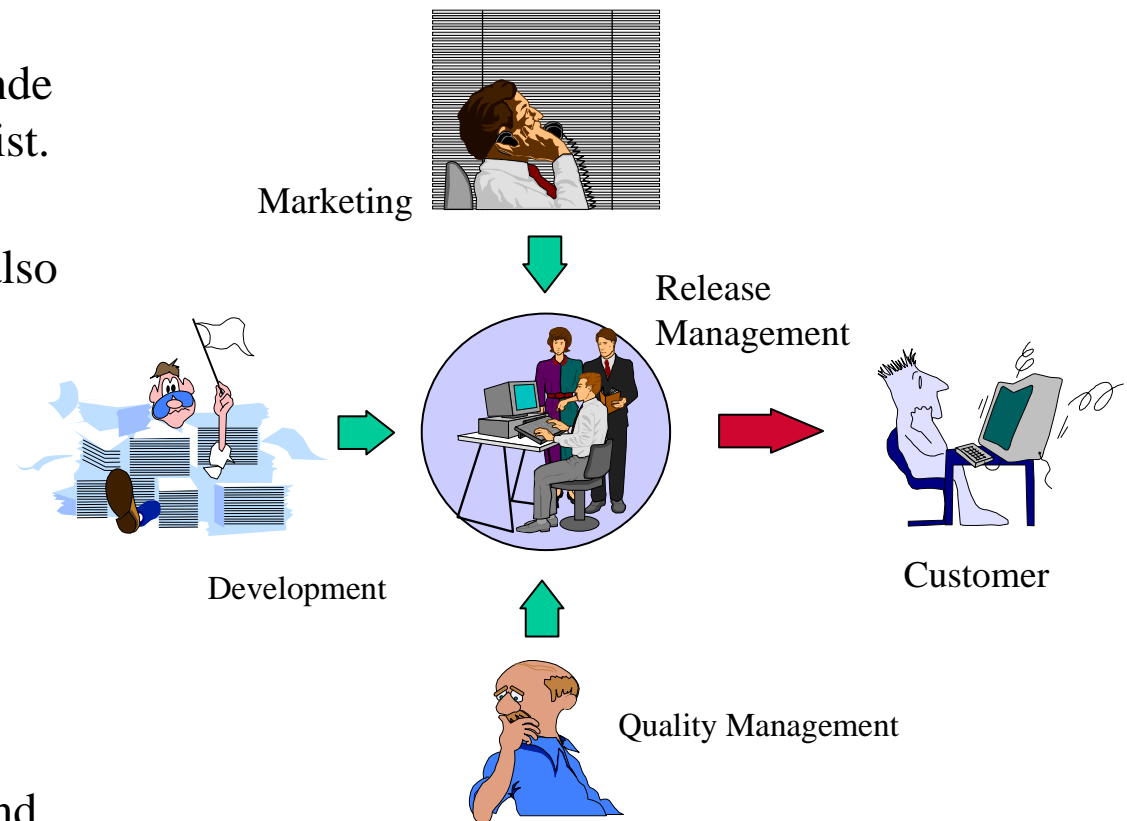
Vorlesung 10:
Release Management
Dr. Erwin Hoffmann

E-Mail: it-pm@fehcom.de



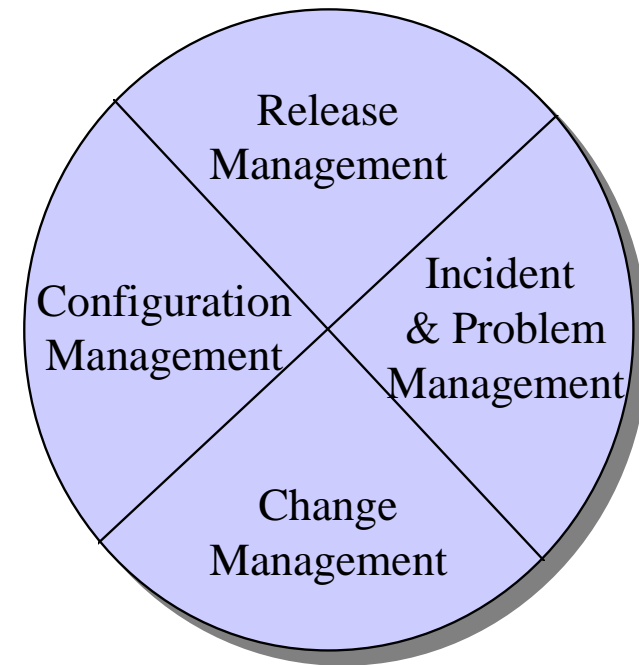
Release und Roll-out Management

- Im Rahmen des IT Projekt-Managements und für ein konkretes Software-Projekt ist das Release Management (RM), diejenige Management Disziplin, von der Kunde bzw. User am direktesten betroffen ist.
- Trotzdem wird diese Management Disziplin von keinem Framework (also weder PMI noch PRINCE2) qualifiziert berücksichtigt.
- Vielmehr wird bestenfalls auf das Release Management innerhalb der ITIL Dienste verwiesen.
- Alternativ wird of Release Management als Teil des Change Managements betrachtet, was aber nicht alle Spezifika des RM genügend eingeht.



Release Management a la ITIL

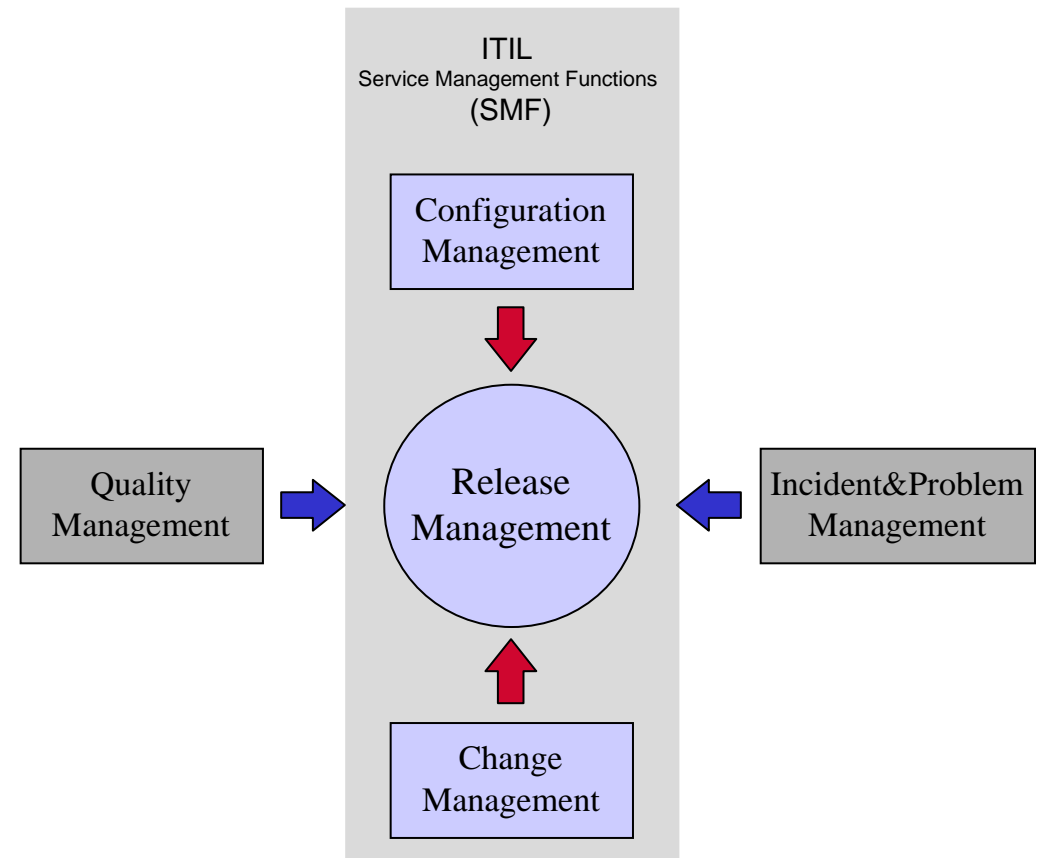
- Das klassische Release Management ist Teil der ITIL (IT Infrastructure Library), genauer ein Bestandteil der *Service Management Funktionen SMF*.
- Im Rahmen des ITIL Service Modells wird Release Management benötigt für
 - die Einführung von umfangreicher bzw. kritischer Hardware-Änderungen.
 - das Roll-Out von Software
 - beim Bündeln komplexer und verknüpfter Änderungen.
- Das Release Management hat hierbei die Aufgabe der Koordination der unterschiedliche Service Provider (z.B. der Software-Entwicklung), der Lieferfirmen und anderer beteiligten Parteien, damit
 - alle Produkte für die Lieferung zur Verfügung stehen
 - die zugehörige Dokumentation vorliegt
 - und ggf. Schulungen angeboten werden.



ITIL Release Management

- Ein zentrales Konzept von ITIL ist die *Definitive Software Library DSL*.
 - Die DSL stellt eine Container für die Programme dar, während
 - die Konfigurationsinformation in der *Configuration Management Database CMDB* gehalten werden.

Es gilt zu beachten, dass die DSL nicht äquivalent zu unserem Software *Version Control System (VCS)* ist; genausowenig wie die CMDB nicht unserem *Document Management System (DMS)* entspricht.
- Der Releaseprozess soll sich nach definierten Vorgaben ausrichten, die niedergelegt werden müssen.
 - Wichtiger Teil dieser Vorgaben ist das Zuordnen der Verantwortlichkeiten und ihre Kompetenzen im Rahmen des Release Managements..



ITIL Release Management (2)

- Bei ITIL werden dem Release Management folgende Hauptaufgaben zugewiesen:
 - *Planung und Überwachung* des Rôle-Outs im Hinblick auf alle geänderten Hard- und Software-Komponenten unter Einschluss der relevanten Dokumente.
 - *Abstimmung mit dem Change Management* bezüglich des Umfangs des Releases und Erstellen eines detailliert Rôle-out Zeitplans.
 - *Verantwortliche Hinterlegung* des neuen Releases mit seinen Komponenten in der CMDB.
 - Erstellung von *kundenspezifischen Release-Dokumenten* und Informationen um die Akzeptanz des neuen Releases zu verbessern und die Erwartungshaltungen geeignet aufzubauen.



ITIL Release Management (3)

- Bezüglich des auszuliefernden Releases benennt ITIL drei unterschiedliche Release-Typen:
 - *Full Release*: Alle Komponenten wurde gemeinsam entwickelt und getestet; sie werden gleichermassen gemeinsam geliefert und implementiert.
 - *Delta Release*: Hierin finden sich spezifische Änderungen und nur bei den Modulen, die seit dem letzten Release angepasst wurden.
 - *Package Release*: Dies ist ein Superset unterschiedlicher Produkte bzw. Releases evtl. unter Einschluss von Third-Party Produkten.



ITIL Release Management (4)

- Das ITIL Model sieht weder explizite Software-Entwicklung noch ein Qualitäts-Management vor und macht lediglich die folgenden Vorschläge:
 - Build Management

Die Soft- bzw. Hardware, die zu erzeugen ist, muss in konsistenter Form für das Release gebaute werden ('build' Prozess),
Die gleichen (build-) Ergebnisse müssen nachvollziehbar erzielt werden und menschliche Fehler sollten ausgeschlossen werden können.
Sobald die Testphase beginnt, ist das Build Management in Verantwortung des Release Managements.
 - Test Prozeduren und Fall-Back Strategien

Jedes Release ist vor seiner Abnahme sorgfältig zu testen und die Kundenakzeptanz ist festzustellen.
Zusätzlich muss ein sog. Fall-back Plan erstellt und dokumentiert werden, der dann in Kraft tritt, falls das Release doch noch fehlerbehaftet ist, bzw. sich nicht installieren lässt.
Die Fall-back Strategien and Pläns sind ebenfalls zu testen und zwar typischerweise in der Referenz-Umgebung.

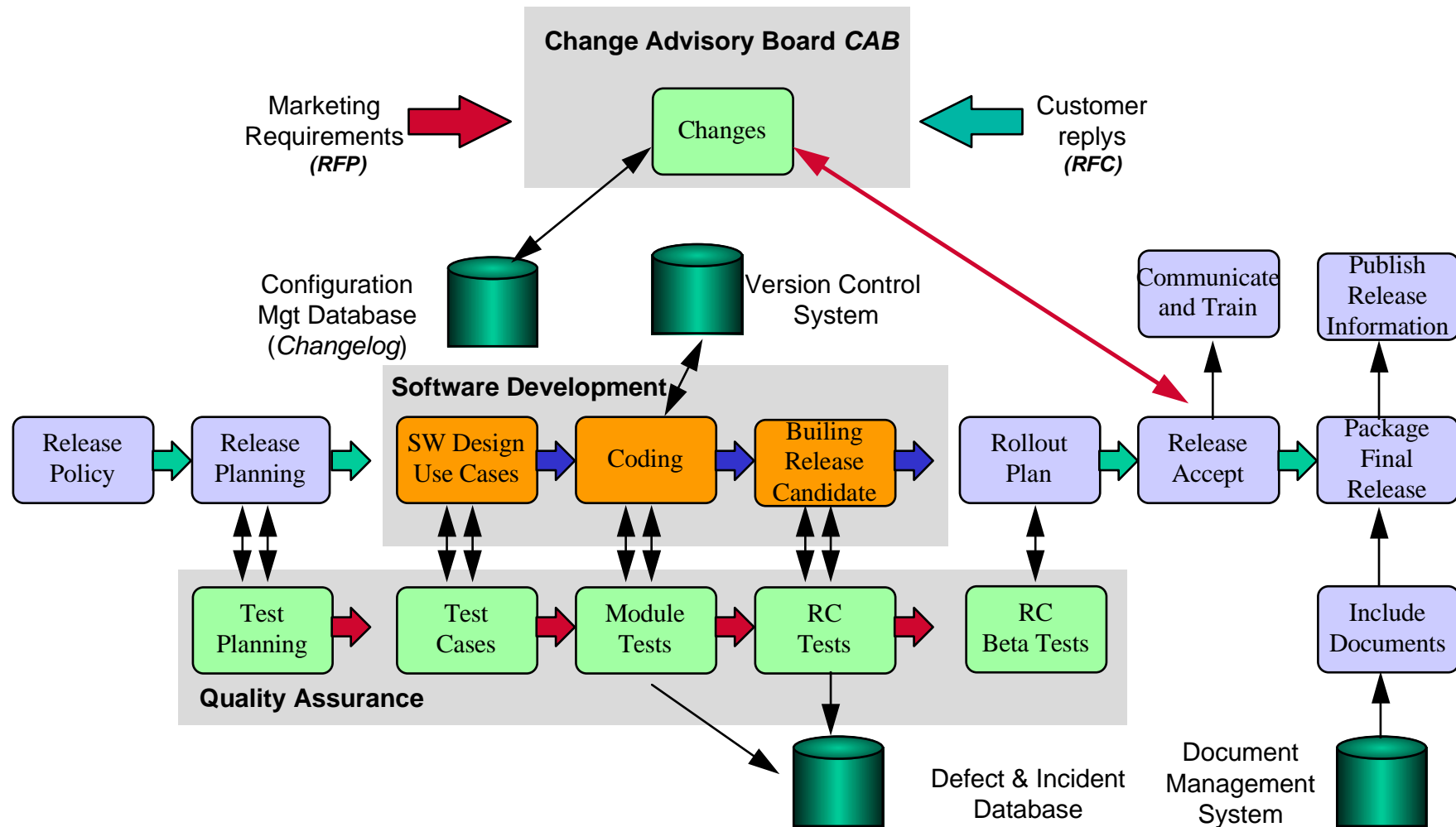


Release Management Überblick

- Alternativ zum ITIL's Ansatz kann im Rahmen des Project Managements
 - Release Management als Teil des Qualität-Management betrachtet werden, da auch hier die Erzielungen einer (Produkt-)Qualität im Vordergrund steht.
- Während das Qualitäts-Management die zentralen Schnittstelle zur Software-Entwicklung aufweist, liegt dies beim Release Management allerdings beim Kunden.
 - Das Release Management ist ebenfalls entscheidender Teil des *Software Delivery Life Cycle SDLC*.



Release Management Überblick (2)



Release Management Institutionen

- Change Advisory Board
 - Der Release Manager ist Mitglied des *Change Advisory Board CAB*.
 - Umfangreichere Releases (z.B. ein Betriebssystem) machen mehrere unterschiedliche Release Manager für die verschiedenen eingeschlossenen Produkte erforderlich.
 - Andere Mitglieder des CAB sind in der Regel die Qualitäts- bzw. Test-Manager, the Verantwortlichen der Software-Entwicklung und möglicherweise auch Vertreter (*bridgeheads*) anderer Firmen, der Produkte im Release aufgenommen sind.
- Software-Entwicklung
 - Die Software-Entwicklung findet aufgrund der im vorherigen Kapitel aufgezeigten Vorgaben statt.
 - Die Koordinierung der SW-Entwicklung der einzelnen Komponenten für das aktuelle Release, ist Aufgabe des Release Managers.
 - Üblicherweise wird die Pflege des Versions-Kontroll-Systems VCS von einer Person vorgenommen.
 - Der letzte Schritt der Entwicklung ist das sog. *Labeling* der einzelnen Module und Komponenten im VCS.
 - Release Manager und die Verantwortlichen der Software Entwicklung haben die gemeinsame Aufgabe, für die Konsistenz und die Vollständigkeit aller Komponenten des Releases zu sorgen.



Release Management Institutionen (2)

- Qualitäts-Sicherung (QA)
 - Der Löwenanteil des QA Prozesses findet während der Phase der eigentlichen Software-Entwicklung statt.
 - Somit trägt der Qualitäts-Manager die Hauptlast sicherzustellen, dass das aktuelle Release weitgehend fehlerfrei ist.
 - Da jedoch kaum eine 100%ige Fehlerfreiheit gewährleistet werden kann, muss der Qualitäts-Manager im Vorfeld eines zu erwartenden Releases Test-Kapazitäten zur Unterstützung eines anschließenden Post-Release Bug-fixing bereitstellen.
 - Soll die Software später auf vielen unterschiedlichen Systemen eingesetzt werden, muss dies bereits in der Testphase berücksichtigt werden.

In diesem Fall kann die Software Entwicklung und das Qualitäts-Management entscheiden, zusätzlich kontrollierte 'Feldtests' (Beta-Test) in Ergänzung zu den In-Haus (Alpha-Tests) vorzunehmen.



Release-Planung

- Die Release-Planung bezieht sich sowohl auf das initiale Release als auch die weiteren Releases.

Rôle	Context	Release Entry Point	Release request type
Senior Executive	Discussions of largest grained needs.		Usually personal interactions and unstructured docuents (emails, presentations, etc.).
	'Final association of most serious issues ideation of exporation.		Request for information (RFI).
	Requests for major new systems.	IT CRM	Starts with personal interactions and moves into process driven.
Unit Executive	Requests for major new systems.		RFP - Request for Proposal.
	Large project level.		
	Requests for new systems, additonal functionality		RFC (high level) Request for Change
	Large project level.	Demand Management System	Starts as process-driven demand request or RFC
Functional area owner		Demand Management System	Starts as process driven demand request or RFC.
	IT Service Request System	Development requirements.	
User	Requests for orderable IT Services Reporting of Incidents	Incident Management System	Process driven for service requests, reporting requests, Incident reports, or Change RFCs (operational)



Initial-Release

- Bereits vor dem Kodieren der ersten Programmzeilen sollte das Release Management festgelegt haben, die die Releases identifiziert und im VCS realisiert werden.
- Die Wahl des VCS und die hiermit einhergehenden Eigenschaften bestimmen zu einem Gutteil die Flexibilität des Release-Prozesses.
 - Die konsistente Benennung der Releases kann mitunter schwierig und auch von Zufälligkeiten geprägt sein.
 - Ursprüngliche stehen folgende Informationen zur Verfügung:
 - Die Versionen der unterschiedlichen Software Module; entweder über das VCS vergeben oder durch den Software-Autor selbst.
 - Die Versionen und möglicherweise Releases zusätzlicher externer Komponenten, die mit dem jetzigen Release gebündelt werden sollen.
 - Die sog. *Build Number* des VCS, die eine bestimmte Konfiguration identifiziert oder ein Schrittzähler für die Anzahl der Kompilierungs/Linker-Versuche zur Generierung der ausführbaren Module darstellt.



Initial-Release (2)

- Zur effizienten Defekt-Nachverfolgung müssen diese Informationen in der Configuration Management Database CMDB aufgenommen und in ihrerer Gesamtheit nun als "Release" bezeichnet werden.
- Um den SDLC zu unterstützen, wird typischerweise von folgendem hierarchischen System Gebrauch gemacht:

- *Major Release*
- *Minor Release*
- *Fixlevel*

und eine spezifische Nummer, z.B. 2.7.18 wird dem Release zugewiesen.

Wie bereits diskutiert kann ein Minor Release als Delta Release aufgesetzt werden, und lässt sich dann ausschliessliche gegen ein spezielles Major Release installieren.

Manchmal wird dies auch als V.R.F Schema bezeichnet, mit V = Version, R = Release und F = Fixlevel.

Auf diese Art und Weise können die logischen Abhängigkeiten der Software-Entwicklung in der Release-Benennung berücksichtigt werden.

- *Delivery Information*

In einigen Fällen wird auch (zusätzlich) die 'Liefernummer' mit angegeben, wobei diese durch Strich "-" oder ein Unterstrich "_" anschliessend an die Release-Nummer angehängen wird.

- *Build Number*

Heute ist es üblich, auch die Build Number in der Release Information mit aufzunehmen. (beispielsweise beim Windows NT, während der in der Öffentlichkeit bekannte Name die Entwicklung bezeichnet.

Andere Firmen nutzen zusätzlich ansprechende Namen für ein spezifisches Release (Apple beim OS X - Panther: 10.3, Tiger: 10.4, Leopard 10.5).



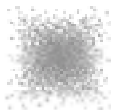
Weitere Releases

- Die Release-Planung für die weiteren Releases hängt von vier Faktoren ab:
 - Dem *Markt*: Das Marketing hat dem Rahmenbedingungen vorzugeben und die 'Trends' zu identifizieren, die es zu folgen gilt.
 - Den *Kunden*: Die Kunden berichten i.d.R. über aufgetretene Fehler und wünschenswerte Erweiterungen.
 - Dem *Qualitäts-Management*: Die offenen Fehler des Vorgänger-Releases sind zu beheben.
 - Der *technische Fortschritt (Entwicklung)*: Geänderte Standards sind im nächsten Release zu berücksichtigen.
- Es existieren zwei Regelprozesse für die Release-Planung:
 - *RFP - Request For Proposal*
in die typischerweise die ursprünglichen Definitionen für zusätzliche Anforderungen einfließen.
 - *RFC - Request for Change*
die vom Change Manager eingestellt und vom Change Advisory Board CAB bestätigt werden.
Alle umgesetzten Änderungen sind zu dokumentieren.
Vor der nächsten Release-Runde muss das Change Advisory Board die vorliegenden Änderungen zu begutachten und diejenigen Auswählen, die in die nächsten Releases einfließen.
Die Änderungen werden nach Modul und Auswirkung charakterisiert, sowie nach Arbeitsumfang bewertet und letztlich priorisiert.



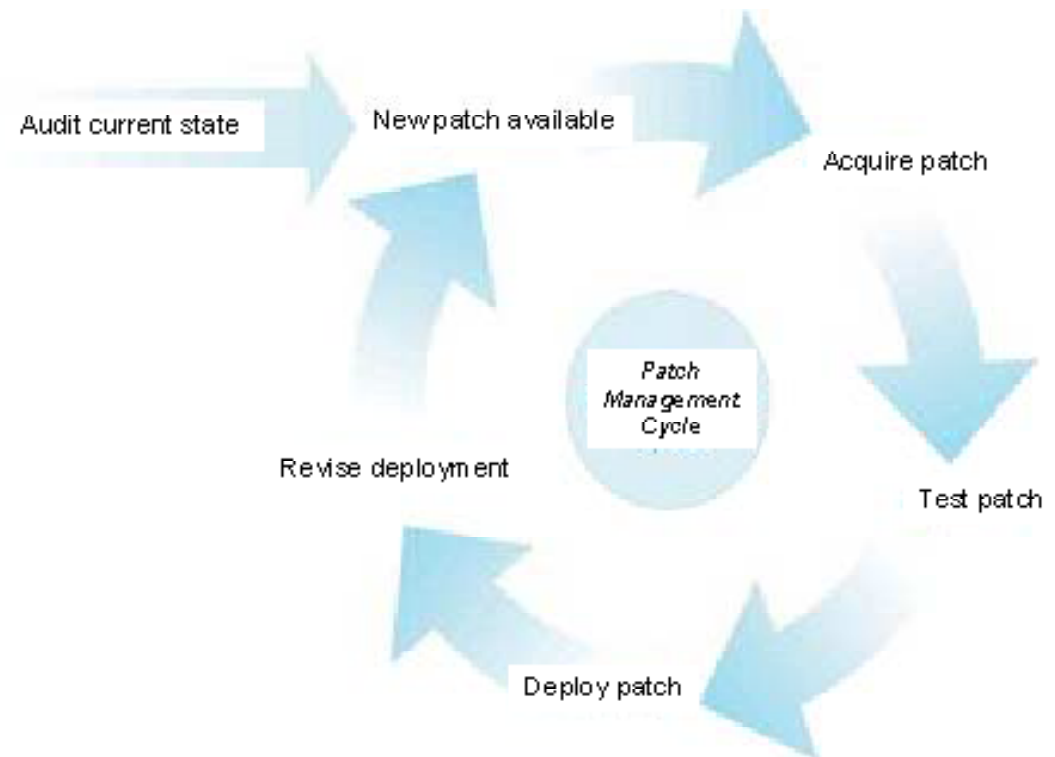
Programm-Code und Patch Management

- Programm-Code und Patch-Management ist Aufgabe der Software-Entwicklung, die allerdings während der Build-Phase durch das Release Management überwacht werden sollte.
- In der Regel wird der Programm-Code vom Software-Entwickler nach eigenem Gutdünken überarbeitet.
 - Besonders bei Open Source Projekten erhält das Projekt zusätzlich Programmmodifikationen externer Entwicklern oder Benutzer.
 - Weiterhin können die Ingenieure Vor-Ort Code-Änderungen durchführen, um ein spezifisches Kundenproblem zu lösen.
- Als *Patch* bezeichnet man 'kleinere' (und kontrollierte) Änderungen am Source-Code, aber auch den ausführbaren Modulen.
- Wird das Patch gegen ein ausführbares Programm durchgeführt, muss es re-engineered und in den Source-Code überführt werden.



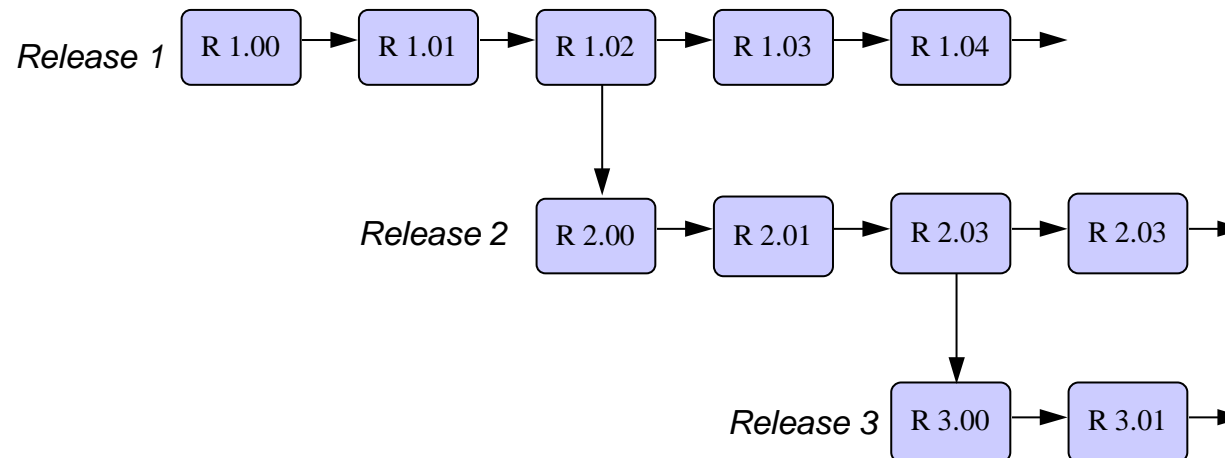
Programm-Code und Patch Management (2)

- Änderungen am Programm-Code und Patche, die die Funktionsweise eines Moduls ändern (und so z.B. einen Fehler beheben) sollten sich folgender Fragen bewusst sein:
 - Worin lag die Ursache für die Programmänderung ?
 - Wer ist für die Pflege des Prgramms zuständig ?
 - Gegenüber welcher Version der Software muss die Änderung vorgenommen werden, und in welches Release sollte die Änderung einfliessen?
 - Hat die Änderung Auswirkungen auf Use und/oder Test Cases ?



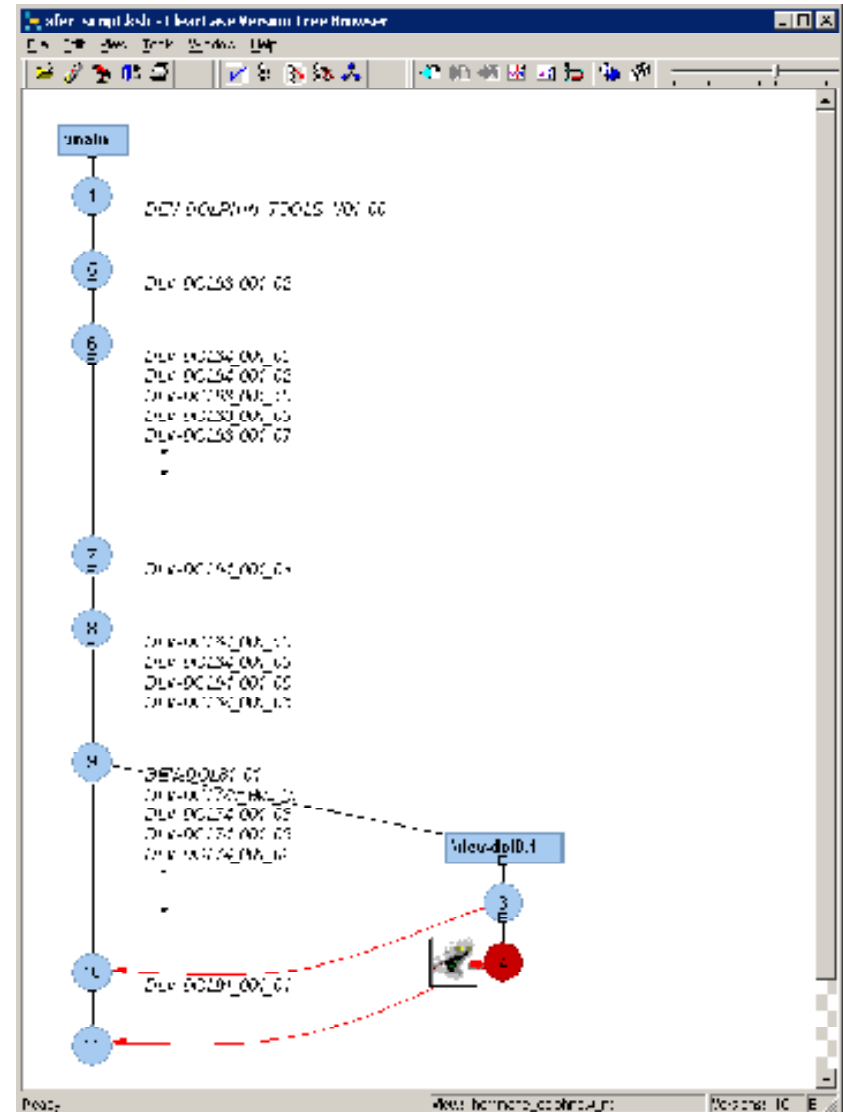
Der Build Prozess

- Die Weiterführung des Programm-Codes wird durch die Entwickler im Rahmen des Release-Plans vorgenommen.
- Innerhalb des VCS besitzen die Entwickler sogenannte 'Views' für bestimmte Releases. Software Modules werden für ein Release 'ein-' und 'ausge-checked'.
- Zwei unterschiedliche Betrachtungsweisen müssen berücksichtigt werden:
 - Der **Branch** - ist das logische Repository der Software im VCS, der gemeinsam für alle Nutzer existiert.
 - Ein **View** - ist der Arbeitsbereich des Entwicklers, wobei ein View sich über mehrere Branches erstrecken kann.
- An das Release Management wird die Anforderung gestellt festzulegen, wann eine neue Verzweigung angelegt wird (*fork, branching*) und somit ein neues Release ansteht.



Build Process (2)

- Branching
 - Im Rahmen des Release Managements ist festzuhalten, das jedes Release auf einem eigenen Branch gepflegt wird, der sein eigenes SDLC besitzt. Daher sollte zu häufiges Branching vermieden werden.
 - Ein unklares Branching führt dazu, dass die Entwickler Programm-Änderungen im falschen Branch vornehmen, da diese (übergreifende) Information in der Regel nicht sichtbar ist.
 - Es ist die Aufgabe des Release Managers oder des Verantwortlichen der Software Entwicklung, das Release-Schedule mit den Entwicklern abzusprechen, um solche Fehler zu vermeiden.
- Labeling
 - Die Software-Komponenten, die zu einem Release gehören werden *ge-labeled*. Es können auch solche Mode gelabeled werden, die zu unterschiedlichen Branches gehören.



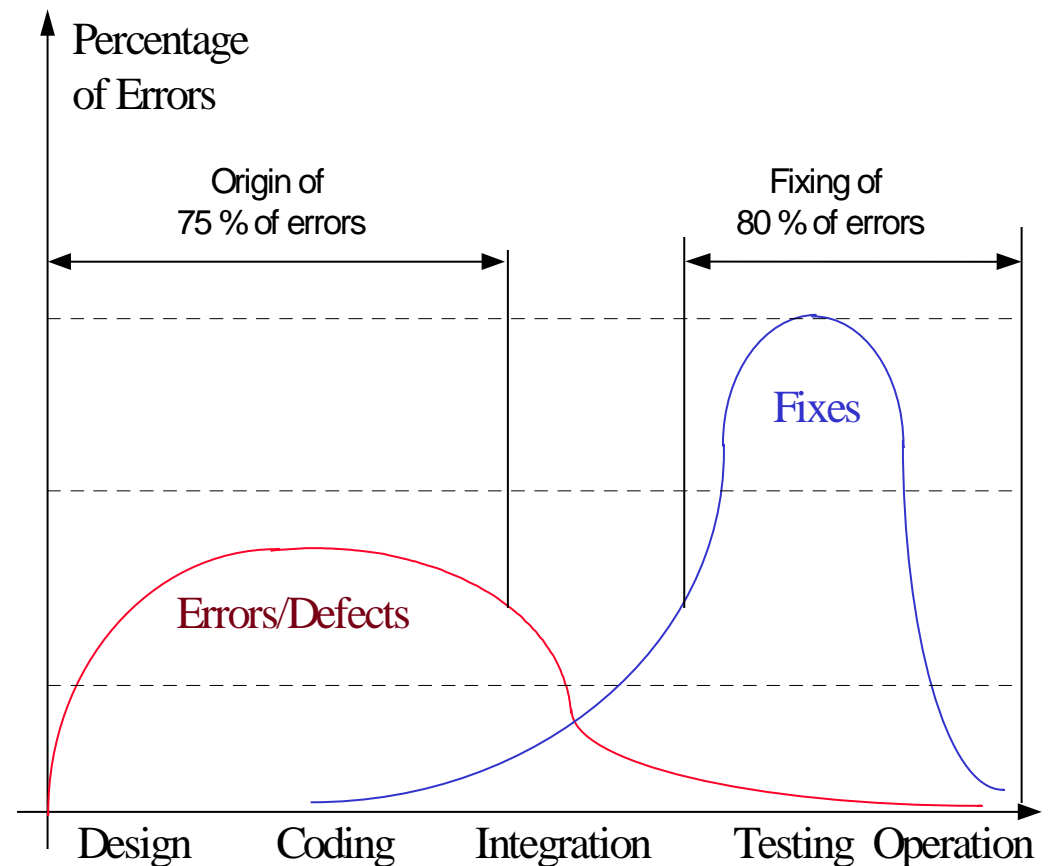
Qualitäts-Sicherung (QA)

- Der Qualitäts-Manager und der Release-Manager müssen für die noch zu behandelnden Defekte im aktuellen Release vorbereitet sein.
 - Bei einem qualifizierten QA System, können die meisten Fehler vor dem Release-Build und der Inbetriebnahme eliminiert werden.
 - Üblicherweise werden die erkannten Fehler als Fixes für das nächste (minor) Release vorgesehen.



Quality Assurance - Hotfixes

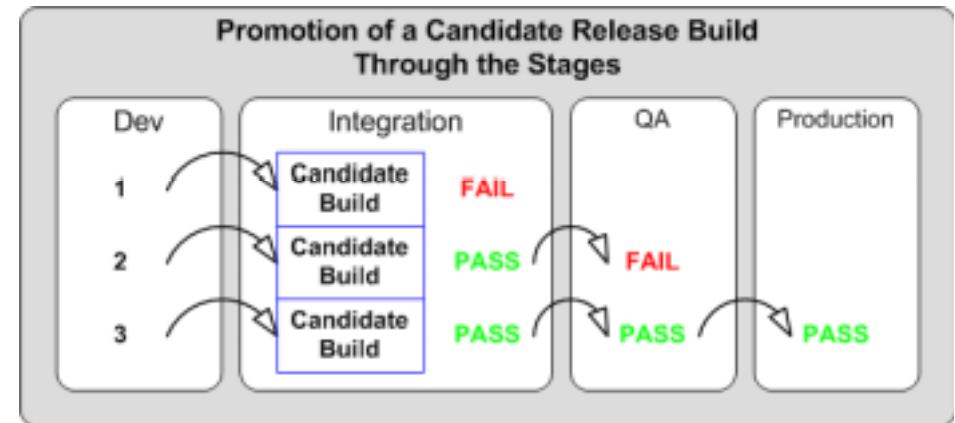
- Unter Umständen ist es notwendig, wichtige Fehler-Fixes in einem eigenen Hotfix-Release unterzubringen.
- Der Terminus 'Hotfix' kann unterschiedliche Bedeutung aufweisen:
 - Gelegentlich werden diese im Rahmen des Standard Release-Zyklus betrachtet, oder aber als tatsächliche "hot-fixes", die für spezifische Kundenprobleme herangezogen werden.
 - Wichtig ist in jedem Fall die Berücksichtigung des Hotfixes im Rahme der allgemeinen Software-Pflege.
 - Üblicherweise sind Hotfixes 'alleinstehend'; verlangen also nicht ausdrücklich die Verkettung innerhalb des SDLC in Bezug auf ein bestimmten Release-Stand.



Release Readiness

- Die Mitteilung der *Release Readiness* (*Freigabe*) ist die Verantwortlichkeit des Release Managers
- Die eigentliche Freigabe liegt jedoch in den Händen des *Change Advisory Board*.
- Abhängig vom Stand des Releases, kann das CAB entscheiden zunächst einen *Release Candidate (RC)* vor dem eigentlichen (final) Release herauszugeben.
- Der Release Candidate kann zur Verfügung gestellt werden:
 - der QA Abteilung
 - der Öffentlichkeit

Die QA Abteilung kann die 'Readiness' im Rahmen ihrer Umgebung feststellen; was natürlich für einen 'public RC' nicht gilt, der jedoch Probleme und Design-Schwächen im realen Umfeld erkennen lassen kann.



Die Auslieferung (Roll-Out Prozess)

- Die Auslieferung kann unterschiedliche Software- und Hardware-Produkte umfassen.

Beispielsweise beinhalten heute Betriebssysteme eine Vielfalt von eigenständigen Komponenten, wie E-Mail-Clients; Web-Browser, Text-Editoren, multi-media Applikationen und andere 'Gadgets'.

- Der Roll-Out Prozess als Teil des Release Managements muss daher unterschiedliche Lieferungen bündeln.

- Im Rahmen des Roll-Out sind die notwendigen Dokumente bereitzustellen und auszuliefern; was abhängig vom Produkt und den Kundengruppe bedeutet:

Installations-Anweisungen

Release-Dokumentation

Benutzer-Handbücher

Administrations-Manuals

- Ist das Roll-Out vorbereitet, heisst das auch, dass die notwendigen Trägermedien zur Verfügung stehen. Waren früher häufig Disketten und CDs dafür notwendig, ist heute vor allem die Kapazität der Download-Server von grossem Belang.
- In jedem Fall liegt es in der Verantwortung des Release Managers, dafür zu Sorgen, dass die notwendigen Ressourcen zeitgerecht und ausreichend zur Verfügung stehen.

