

Studieninformation zum Gebiet „Software Engineering“

Beschreibung des Gebietes

Warum ist die Erstellung großer Softwaresysteme immer noch so fehleranfällig? Über 20% der industriellen Softwareentwicklungsprojekte werden vorzeitig abgebrochen. Über 30% überziehen Kosten und Zeit dramatisch.

Das Gebiet Software Engineering stellt *Beschreibungstechniken, Methoden, ingenieurmäßige Prinzipien und Werkzeuge* bereit, die dazu dienen die Entwicklung beherrschbar zu machen.

Im Zentrum stehen dabei die folgenden Aspekte:

Komplexität

Software ist groß und immateriell. Anwendungen entstehen in einem komplexen Umfeld von Plattformen und Middleware, der Code umfasst manchmal zig Millionen Zeilen (z.B. SAP R/3). Software Engineering ermöglicht die systematische Zerlegung in und Integration von Softwarekomponenten. Software Engineering ermöglicht die Beschreibung (z.B. Unified Modeling Language **<<UML>>**) und die Integration verschiedener Sichten auf den Code.

Gestaltung von Arbeit und Technik

Software wird von Menschen genutzt. Ihre Anwendung macht nur dann Spaß und bringt den erhofften (zeitlichen oder finanziellen) Gewinn, wenn sie auf die Bedürfnisse der Menschen zugeschnitten ist. Dazu muss man nicht nur die Technologie, sondern auch die Interessen (Anforderungen) der Menschen verstehen, und die Technologie im Sinne dieser Interessen gestalten. Software Engineering integriert moderne Methoden des Wissenserwerbs, der Verhandlungsführung, und der Entscheidungsunterstützung in den Entwicklungsprozess.



Abbildung 1: Wissen, Verhandlung, Entscheidung

Technologiewandel

Software ist Teil komplexerer Produkte. Diese Produkte sind einem schnellen Wandel unterworfen. Software Engineering stellt Methoden der Wiederverwendung und des Konfigurationsmanagements zur Verfügung, die eine Produktpassung in sehr kurzen Zyklen ermöglicht. Für den Markterfolg von Software ist eine an modernen Managementmethoden ausgerichtete Marktstrategie notwendig.

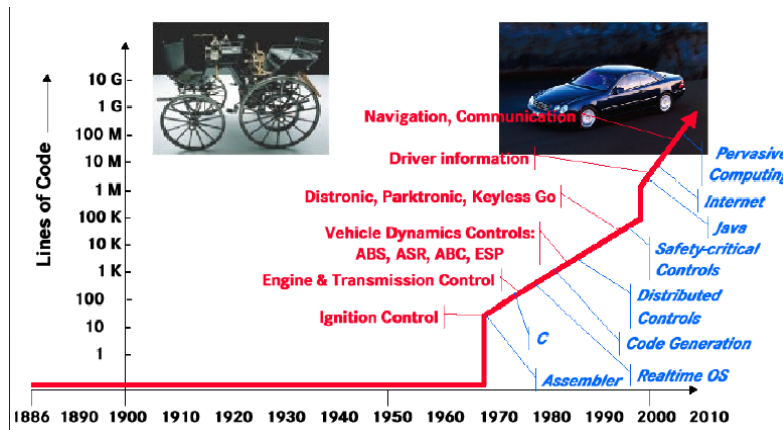


Abbildung 2: Software im Automobil (DaimlerChrysler)

Teamarbeit

Software wird im Team entwickelt, manchmal sind Hunderte von Personen über viele Jahre daran beteiligt (z.B. in der Luft- und Raumfahrt), oft sind die Personen über viele Standorte verteilt. Software Engineering unterstützt die Arbeitsorganisation und die Bildung eines gemeinsamen Verständnisses durch moderne Methoden der Kommunikation, des Qualitäts- und Projektmanagements, z.B. Workflowsysteme und Groupware, Videokonferenzen, Wissens- und Risikomanagement.

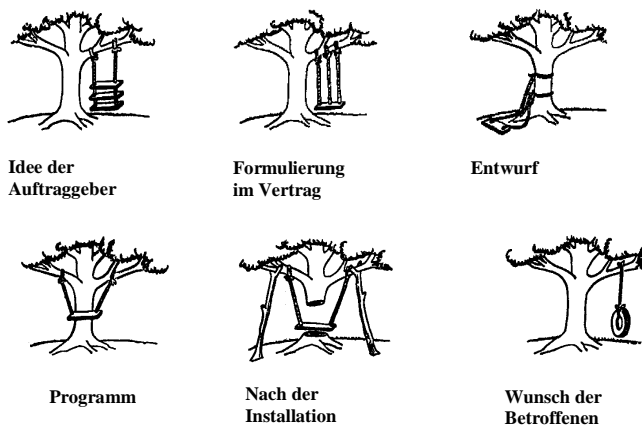


Abbildung 3: Kommunikationsprobleme in der Entwicklung

Typische Fragestellung in Forschung und Entwicklung

Entwurf einer geeigneten Architektur

Für eine über das Web benutzbare Anwendung muss eine geeignete Architektur definiert werden. Die Datenverbindung über das Netz darf nicht zu langen Antwortzeiten oder Übertragungsfehlern führen.

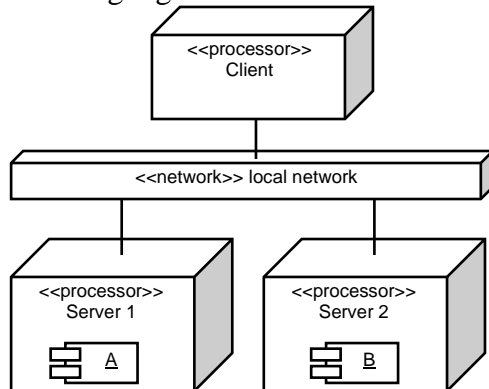


Abbildung 4: Architektur

Definition der Anforderungen

In einer Lebensmittelkette soll ein neues Logistiksystem erstellt werden. Dazu sind mit allen Beteiligten die Anforderungen an das System zu definieren. Dies ist nur möglich, wenn man die Arbeitsprozesse der Benutzer verstehen und beschreiben kann.

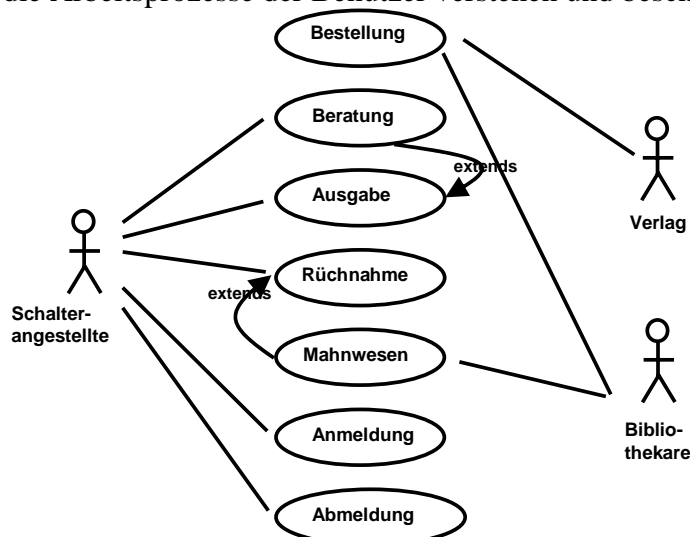


Abbildung 5: Aufgabenbeschreibung

Änderungsmanagement

Anforderungsspezifikationen von komplexen Systemen, z.B. einem neuen Autotyp, füllen oft mehrere Ordner. Diese Daten kann man heutzutage in einem Anforderungsmanagementwerkzeug verwalten. Wie kann so ein Werkzeug Änderungen an Modellen und Text optimal unterstützen?

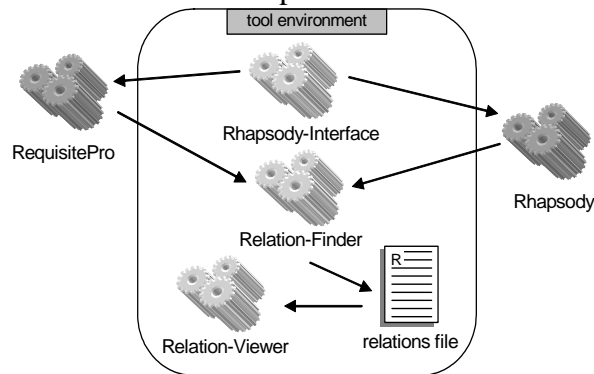


Abbildung 6: Konzept zur Verbindung von Modellierungs- und Managementwerkzeug

Testen mobiler Anwendungen (Projekt MORABIT)

Die Flexibilität mobiler Anwendungen ist schon in naher Zukunft für Unternehmen unverzichtbar. Die Herausforderung für die Akzeptanz mobiler Softwarelösungen sind die mit der Flexibilität verbundenen Zuverlässigkeitsrisiken. Traditionelle Qualitätsmanagement-Verfahren bieten hierfür aus zwei Gründen keine Lösung:

- (1) sie sind nur zur Entwicklungszeit anwendbar und werden so dem ständig wechselnden Kontext mobiler Komponenten nicht gerecht.
- (2) sie berücksichtigen nicht die Ressourcen-Beschränktheit mobiler Systeme (z.B. verfügbare Rechenleistung, Arbeits- und Festplattenspeicher, Zugang zu Netzwerken, die vorhandenen Ausgabemodalitäten, insbesondere Größe und Farbtiefe des Bildschirms).

Deshalb sind zeitgleich zur Entwicklung mobiler Technologie auch Strategien und Werkzeuge für das Qualitätsmanagement mobiler Softwarelösungen zur Verfügung zu stellen. Die zugrunde liegende Forschungsfrage ist, wie Unternehmen angesichts der Dynamik und Ressourcen-Beschränktheit mobiler Anwendungen das Qualitätsmanagement, insbesondere die Planung, Spezifikation und Durchführung von Tests, effizient gestalten können.



Abbildung 7: Mobile Anwendungen