Horst Lichter, Martin Glinz (Hrsg.) (2001): Software Engineering im Unterricht der Hochschulen SEUH 7 – Zürich, dpunkt.verlag, Heidelberg.

Erfahrungen mit einem Werkzeug zur Projektunterstützung

Andreas Spillner
Hochschule Bremen
Fachbereich Elektrotechnik und Informatik
Neustadtswall 30
D-28199 Bremen
spillner@informatik.hs-bremen.de

Zusammenfassung

Der Beitrag berichtet über Erfahrungen beim Einsatz des ProVista® method kit [1] in einem studentischen Ausbildungsprojekt im Fachbereich Elektrotechnik und Informatik der Hochschule Bremen.

ProVista® method kit ist von der Firma 3soft in Erlangen [2] entwickelt worden und beschreibt einen Software-Entwicklungsprozess. Es ist kein Werkzeug im üblichen Sinne, sondern eher eine Sammlung von nützlichen Hinweisen, Verfahren und Dokumenten. Es ist nach dem Baukastenprinzip aufgebaut, so dass die zentralen Elemente (Methoden, Phasen, Rollen und Dokumente) leicht auf die jeweiligen Projekterfordernisse und Anwendungsgebiete anpassbar, kombinierbar und erweiterbar sind. Die Darstellung erfolgt auf Basis der World-Wide-Web-Technologie.

Neben den Beschreibungen und Erörterungen sind Dokumentmuster direkt verfügbar (in MS-Office-Formaten), die sich sehr gut als Vorlagen für die zu erstellenden Dokumente eignen.

Durch das ProVista® method kit konnte das studentische Projekt, von dem hier berichtet wird, einen Rahmen zur Unterstützung der Projektabwicklung finden, ohne das es zu langwierigen Diskussionen über eher unwichtige Fragen kam.

1 Einleitung

Die Ausbildung in Softwaretechnik sollte insbesondere an einer Fachhochschule sehr praxisnah und Ingenieur-gemäß erfolgen. Eine Schwierigkeit bei der Vermittlung des Lehrinhalts besteht oft darin, dass die Studierenden den Praxisbezug nicht direkt erkennen. Meist haben sie auch aus ihrer Tätigkeit neben dem Studium andere Vorgehensweisen - meist keine - bei der Softwareentwicklung kennengelernt. Es gilt, einen strukturierten Entwicklungsprozess zu vermitteln und nachzuweisen, dass dieser auch in der Praxis angewendet wird.

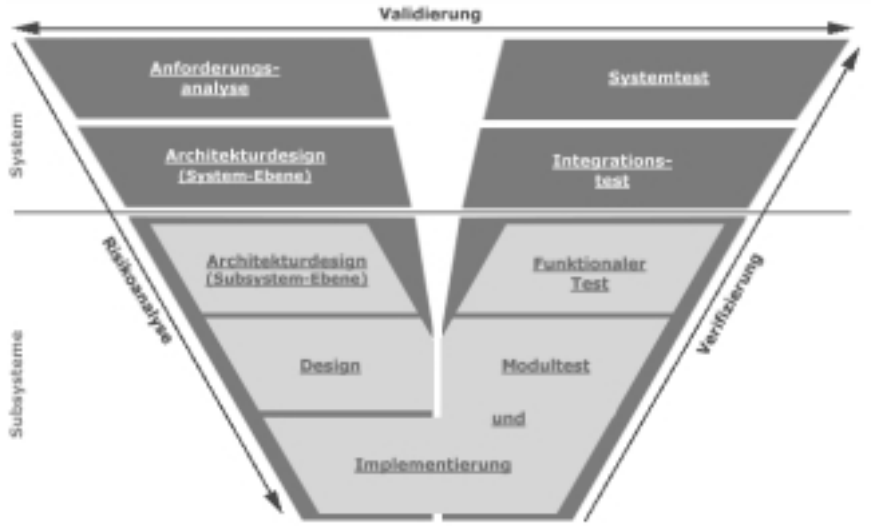

Abb. 1: Phasenmodell

Das in der Lehre verwendete Werkzeug ProVista® method kit ist ein Produkt, welches von der Firma 3soft vertrieben wird und in mehreren Firmen (ca. 50, Stand Okt. 2000) nach einer entsprechenden Anpassung an deren Bedürfnisse eingesetzt wird. Den Studierenden wird durch das Werkzeug eine Vorgehensweise der Softwareentwicklung vermittelt, die auch in der Praxis ihre Anwendung findet.

Studierende wurden auf dem SQM-Kongress 1998 in Köln auf das Werkzeug aufmerksam. Nach der Kontaktaufnahme mit der Firma 3soft wurde der Hochschule eine Version für Lehrzwecke zur Verfügung gestellt. Ein Vergleich mit anderen Werkzeugen zur Unterstützung des Software-Entwicklungsprozessese fand nicht statt.

2 ProVista® method kit

"ProVista® method kit dient der umfassenden Definition von Software-Entwicklungsprozessen und geht nach dem Baukastenprinzip vor. ProVista® method kit beschreibt alle Phasen des Entwicklungsprozesses, verteilt Rollen und Aufgaben, stellt standardisierte Dokumentenvorlagen und Checklisten zur Verfügung, benennt und erklärt alle notwendigen Entwicklungsmethoden. Diese Elemente werden nicht wie sonst üblich in Papierform, sondern auf Basis der World-Wide-Web-Technologie im HTML-Format präsentiert." [1]

Das ProVista® method kit besteht aus vier Hauptteilen, die allerdings untereinander in enger Verbindung stehen. Es gibt Phasen, Rollen, Methoden und Dokumente. Die vier Teile sollen im folgenden kurz vorgestellt werden.

2.1 Phasen

Die Methoden und Dokumente werden in ein Phasenmodell eingeordnet. Das Modell (s. Abb. 1) orientiert sich an dem V-Modell und wird von 3soft selbst für ihre Entwicklungen eingesetzt. Es werden die Phasen Anforderungsanalyse, Architekturdesign auf System- und Sub-System-Ebene, Design auf Modul-Ebene und die Implementierung unterschieden. Beim Test wird zwischen Modul-, Funktions-, Integrations- und Systemtest-Phasen differenziert.

Abb. 2: Übersicht zur Phase Anforderungsanalyse

Ausführliche Beschreibungen zu allen Phasen lassen sich durch Anklicken der entsprechenden Verweise erreichen. Zu jeder Phase sind, neben einer kurzen Beschreibung der Ziele, die Ein- und Austrittskriterien genannt sowie die bei dieser Phase erstellten Arbeitsergebnisse zusammen mit den einzusetzenden Methoden. Abbildung 2 zeigt die Übersicht der Phase Anforderungsanalyse. Die weiterführenden Informationen sind über die Verweise erreichbar.

2.2 Rollen

In der Softwareentwicklung sind unterschiedliche Aufgaben zu erledigen, die bestimmten Rollen zugeordnet werden können. Im ProVista® method kit sind zehn unterschiedliche Rollen benannt. Zu jeder Rolle gibt es eine Kurzbeschreibung der Aufgabe, Vorgaben, die erfüllt sein müssen, die Beschreibung der Aktivitäten und eine Liste der Dokumente, die für diese Rolle relevant sind.

Für die Rolle "Entwickler" gibt es folgende Beschreibung mit der entsprechenden Möglichkeit, direkt weitere Informationen abzurufen:

Kurzbeschreibung
Der Entwickler bearbeitet einen Teilbereich (z.B. ein oder mehrere Module, Subsystem, ...) und ist dort insbesondere für das detaillierte Design und die anschließende Realisierung dessen zuständig.

Vorgaben
Der Entwickler erhält
- den Anforderungskatalog für seinen Teilbereich in Form eines Pflichtenheftes,
- Terminvorgaben aus dem Projektplan.

Aktivitäten
Der Entwickler
- erstellt nach einer entsprechenden Analyse ein Design für seinen Teilbereich,
- modelliert unter möglichst weitgehender Wiederverwendung existierender Module,
- führt die Implementierung der Module durch und führt auch erste Modultests durch,
- fügt die im Rahmen seiner Tätigkeit entstandenen Dokumente der Dokumentation für das Gesamtprojekt hinzu,
- informiert den Projektleiter regelmäßig über den Stand seiner Arbeiten.

Dokumente
Der Entwickler ist verantwortlich für
- Moduldesign
- Quellcode

Er arbeitet bei der Erstellung der folgenden Dokumente mit:
- System-Design
- Release Notes

2.3 Methoden

Die im ProVista® method kit beschriebenen Methoden sind wie folgt gegliedert und mit entsprechenden Verweisen versehen (s. Kasten).

Projektmanagement
- Projektplanung
- Berichterstattung
- Meilensteintrendanalyse
- Risikomanagement

Zu jeder Methode gibt es eine Kurzbeschreibung, die Erörterung des Vorgehens, Erläuterungen zur Notation und Richtlinien zur Anwendung der Methode.

- Anforderungsmanagement
  - Anforderungsanalyse
  - Requirement Keys
  - Anforderungsverfolgung
- Design
  - Unified Modeling Language
- Implementierung
  - Programmierrichtlinien
- Integration & Test
  - Integrationsplanung
  - Teststrategien
- Support & Organisation
  - Konfigurationsmanagement
  - Change Management
- Review-Techniken

In der Kurzbeschreibung wird Sinn und Zweck der Methode beschrieben. Im Vorgehen wird detailliert dargelegt, wie und unter welchen Randbedingungen die Methode eingesetzt werden soll. Die Notation wird mit Beispielen veranschaulicht. So sind zum Beispiel bei den Programmierrichtlinien für die Programmiersprachen C, C++ und Assembler jeweils ausführliche Informationen abrufbar.

Als vierter Punkt bei den Methodenbeschreibungen sind Richtlinien angegeben, die konkrete Hinweise zum Gebrauch der jeweiligen Methode geben.

2.4 Dokumente

ProVista® method kit stellt Informationen zu folgenden Dokumenten zur Verfügung (s. Kasten).

- Projektmanagement
  - Projektplan
  - Statusbericht
  - Risikoanalyse
- Anforderungsmanagement
  - Lastenheft
  - Pflichtenheft
- Design
  - System-Designdokument
  - Modul-Designdokument
- Implementierung
  - Quellcode
- Integration & Test
  - Testplan
  - Modul-Testspez./-dokumentation
  - Funktionale Testspez./-doku.
  - Integrations-Testspez./-doku.
  - System-Testspez./-doku.
- Support & Organisation
  - Release-Notes
  - Review-Protokoll

Zu jedem Dokument gibt es jeweils eine Kurzbeschreibung, die einzusetzende Methode und die jeweiligen Verantwortlichen. Daneben sind meist eine Mustervorlage und eine Checkliste abrufbar.

Die Kurzbeschreibung verdeutlicht den Nutzen der Dokumente und gibt auch eine knappe inhaltliche Richtung vor. Die einzusetzende(n) Methoden zur Erstellung der Dokumente werden ebenso beschrieben wie die Rollen, die verantwortlich an der Erstellung der Dokumente beteiligt sind.

Besonders hilfreich bei den Dokumenten sind Mustervorlagen und Checklisten, die bei fast allen Dokumenten zur Verfügung stehen.

Die Mustervorlagen sind in MS-Office-Formaten, umfassen meist eine Standard-Gliederung und enthalten zu den einzelnen Kapiteln Hinweise zu den jeweiligen Inhalten. Die Checklisten sind eine Zusammenstellung von Fragen, die helfen, die Dokumente vollständig zu erstellen.

3 Einsatz im studentischen Ausbildungsprojekt

Im Fachbereich Elektrotechnik und Informatik der Hochschule Bremen haben die Studierenden ein Projekt zu absolvieren, das zwei Semester in Anspruch nimmt (insgesamt 8-12 SWS). Im Projekt entwickeln ca. zehn Studierende einen Prototypen eines Softwaresystems oder eines Systems, das aus Hard- und Software-Anteilen besteht. Dabei wird meist der gesamte Entwicklungsprozess durchlaufen. Im Projekt, das ProVista® method kit eingesetzt hat, wurde ein Editor zur Erstellung von UML-Klassendiagrammen realisiert. Neben der graphischen Darstellung war eine Komponente für die Umsetzung der Diagramme in einen Code-Rahmen zuständig und eine weitere kontrollierte die Einhaltung von Richtlinien in Bezug auf Namensgebung und anderer Festlegungen.

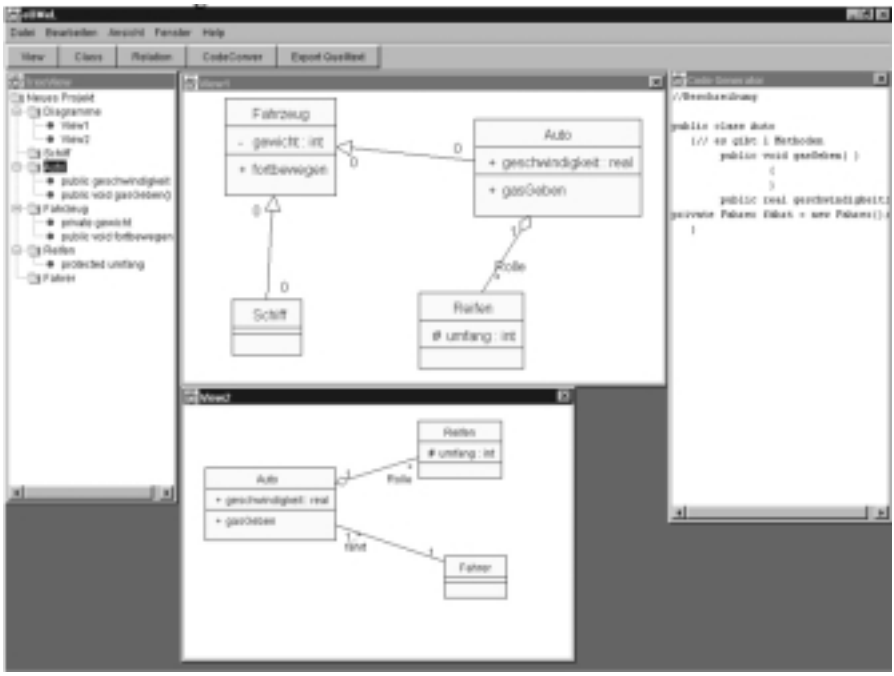

Abb. 3: Graphische Oberfläche des Prototyps

Im Projekt wurde festgelegt, welche Teile des ProVista® method kit sinnvollerweise übernommen und eingesetzt werden können. Es wurde also eine Art Tailo-

Horst Lichter, Martin Glinz (Hrsg.) (2001): Software Engineering im Unterricht der Hochschulen SEUH 7 – Zürich, dpunkt.verlag, Heidelberg.

ring durchgeführt, allerdings eher parallel zum Projektverlauf und nicht durchgängig für alle Phasen zu Beginn des Projektes.

Im folgende werden einige Teile exemplarisch vorgestellt, um die Verwendung des Werkzeugs im Projekt zu veranschaulichen.

3.1 Das Pflichtenheft

Das Pflichtenheft für das Ausbildungsprojekt wurde anhand der Mustervorlage erstellt. Neben der Standardgliederung sind in der Mustervorlage folgende Angaben für das Deckblatt des Pflichtenheftes zu entnehmen: Titel, Verteiler, Beschreibung, Autor, Co-Autor, Ablageort, Version, Status, Datei und Datum. Da die Gliederung und das Deckblatt des Pflichtenheftes in der Praxis verwendet wird, gab es keinen Diskussionsbedarf über Sinn und Zweck der einzelnen Angaben. Zu den jeweiligen Gliederungspunkten ist in der Mustervorlage angegeben, welche Inhalte unter der Gliederungsüberschrift einzufügen sind.

Unter Punkt 1.3 Allgemeine Beschreibung sind folgende Hinweise angegeben:

> Die Zielsetzung enthält die Erklärung dafür, warum ein neues System zu entwickeln ist. Ggf. kann auch ein IST-Zustand beschrieben werden, der durch die Einführung des neuen Systems verbessert werden soll.
>
> Die Beschreibung sollte enthalten:
>
> - die Festlegung des offiziellen Namens,
> - die Ziele, die mit dem Projekt verfolgt werden,
> - die Einordnung in ein evtl. bestehendes Gesamtkonzept,
> - eine Problembeschreibung,
> - eine Abgrenzung zu evtl. vorhandenen anderen Projekten.

Die einzelnen Punkte wurden entsprechend der Projektvorgabe ausgefüllt. Gliederungspunkte, die für das Projekt nicht sinnvoll erschienen, wurden nach einer kurzen Diskussion weggelassen. Die Erstellung des Pflichtenheftes konnte zügig durchgeführt werden und die Diskussion beschränkte sich auf die inhaltlichen Punkte und Angaben. Eine wenig sinnvolle Diskussion über Layout-Fragen, Reihenfolgen und Aufnahme einzelner Punkte, kam durch die Verwendung der Vorlage gar nicht erst auf.

3.2 Rollen

Der nächste Schritt im Projekt betraf die Verteilung der Rollen. Jeder Studierende hat sich näher mit einer Rolle beschäftigt und die jeweiligen Aufgaben der Rolle bei einem Projekttreffen vorgestellt. Es wurde auf Grundlage der Informationen aus der Softwaretechnik-Vorlesung entschieden, ob die jeweilige Rolle im Projekt sinnvoller Weise durch eine Person besetzt werden sollte.

Die Rollen Designer, Entwickler und Tester sollten von allen Beteiligten wahrgenommen werden, da es sich um ein Projekt handelt, das die Ausbildung und nicht die Produktentwicklung als Schwerpunkt hat. Für die Projektleitung und die Systemadministration wurden jeweils zwei Studierende vorgesehen. Ein Student erhielt die Rolle des Qualitätsmanagers. Die verbleibenden vier Rollen (Technischer Redakteur, Sicherheitsbeauftragter, Change Manager, Konfigurationsmanager) wurden als nicht relevant für das Projekt angesehen. Dabei war die Rolle des Konfigurationsmanagers umstritten. Einige Studierende waren der Meinung, dass diese Rolle und ein unterstützendes Werkzeug notwendig sind. Sie konnten sich allerdings bei einer Abstimmung nicht durchsetzen. Allgemein wurde erwartet, dass nicht so viel unterschiedliche Versionen entstehen werden. Während des Projektes wurden die Rollen nicht gewechselt. So konnten sich die Studierenden intensiv mit einer Rolle vertraut machen und tiefere Kenntnisse erlangen.

3.3 Projektplanung

Im Werkzeug sind die grundlegenden Aktivitäten zur Projektplanung beschrieben. Diese wurden allerdings für das Ausbildungsprojekt auf das Notwendigste reduziert. Es gab eine grobe zeitliche Planung der einzelnen Aktivitäten. Die Einhaltung der festgelegten Bearbeitungszeiträume wurde von der Projektleitung kontrolliert. Eine Anpassung der Planung an die Realität wurde zur Mitte des zweiten Projektsemesters vorgenommen. Die in der ersten Planung freigelassenen zwei Wochen am Ende des Semesters mussten als Projektzeit hinzu genommen werden.

3.4 Reviews

Reviews werden nach ProVista® method kit zur Qualitätssicherung nahezu aller erstellten Dokumente verwendet. Die Studierenden haben das beschriebene Review-Verfahren auf die Projektbedürfnisse angepasst. Ebenso ist mit der Protokoll-Vorlage verfahren worden.

Reviews wurden von den fünf studentischen Arbeitsgruppen am intensivsten in der Codierungsphase eingesetzt. Jeweils eine Gruppe führte ein Review der vorliegenden Dokumente einer anderen Gruppe durch. Die Protokolle dienten der Informationsvermittlung, allerdings gab es zusätzlich eine Reihe von direkten Diskussionen unter den Gruppen. Einhellige Meinung war, dass Reviews sehr sinnvoll sind und zu einer erheblichen Qualitätssteigerung geführt haben.

3.5 Statusberichte und Protokolle

Im Projektmanagement sind Statusberichte der einzelnen Gruppen in bestimmten Zeiträumen vorgesehen. Bei studentischen Projekten gibt es leider oft Schwierigkeiten, den Fortschritt der einzelnen Gruppen nachzuvollziehen. Aussagen wie „Wir haben 80% des Programms fertig“ oder „Wir sind voll im Zeitplan“ sind nur schwer bzw. gar nicht kontrollierbar.

Die Statusberichte haben folgenden Inhalt:

1. Historie
2. Inhaltsverzeichnis
3. Abstrakt
4. Gestellte Ziele im Berichtzeitraum
5. Aktueller Projektstand
6. Probleme/ Störungen / Risiken/ Korrekturen
7. Offene Punkte
8. Stellungnahme des Qualitätsmanagers
9. Ziele bis KW

Durch die Verwendung der Statusberichte waren die oben erwähnten Aussagen nicht mehr möglich.

Es konnte frühzeitig auf sich abzeichnende Probleme reagiert werden. So wurde rechtzeitig vor Ende der Implementierungsphase von der Projektleitung dafür gesorgt, dass Teile der Arbeit einer Gruppe von einer anderen übernommen wurden.

3.6 Phasenmodell

Das im Werkzeug vorgestellte Phasenmodell (s. Abb. 1) wurde nicht angewendet. Das Modell legt eine sequentielle Entwicklung nahe. Diese erschien für das Ausbildungsprojekt nicht angemessen, da frühzeitig eine erste Version des Programms vorliegen soll. Andere Modelle der Softwareentwicklung werden im ProVista® method kit nicht unterstützt. Generell lassen sich aber alle zur Verfügung gestellten Methoden, Dokumentenvorlagen und Rollen auf die Belange eines Projektes anpassen. Das Tailoring wird dann in Kooperation mit dem Kunden von der Firma 3soft durchgeführt. Die entsprechende Anpassung für das Ausbildungsprojekt wurde von den Projektbeteiligten selbst durchgeführt.

Im Projekt wurde inkrementell entwickelt. Grundlage war der Entwicklungsprozess in Iterationen, wie er in [3, Kapitel 4.4] beschrieben wird. Es gab während des Projektes zwei Iterationen, also aufeinander aufbauende Ausbaustufen des Editors. Die erste Iteration wurde allerdings aus Zeitgründen nicht ausgiebig getestet.

4 Bewertung

ProVista® method kit ist sicherlich nicht als Werkzeug im engeren Sinne anzusehen. Es ist eine sehr sinnvolle Zusammenstellung der Phasen, Rollen, Methoden und Dokumente, die in der Softwareentwicklung von Bedeutung sind. Durch die Aufbereitung der Informationen im HTML-Format und die Bereitstellung von Mustervorlagen in MS-Office-Formaten ist keinerlei oder nur eine sehr geringe Einarbeitungszeit notwendig. Die Informationen sind mit jedem Browser abrufbar. Diese Art der Aufbereitung wird von den Studierenden gerne angenommen und viel intensiver genutzt als ein entsprechendes Buch mit gleichem Inhalt.

Im Fachbereich Elektrotechnik und Informatik steht den Studierenden bereits eine Web-basierte Lernplattform [4] zur Verfügung, so dass sie es gewohnt sind, mit

dem Medium zu arbeiten. In dieser Lernplattform wurden auch alle Dokumente des Projektes verwaltet.

Durch das Werkzeug wurden Standards für Vorgehensweisen und Dokumente gesetzt, die von den Studierenden ohne große Diskussion akzeptiert und auch eingehalten wurden. Die Vorlagen sind ohne Aufwand auf die jeweilige Projektsituation anpassbar, was nahezu für alle Dokumente vorgenommen wurde und die Akzeptanz insgesamt weiter erhöhte. Die Studierenden hatten dadurch nie das Gefühl, unnütze Dokumente erstellen zu müssen; im Gegenteil, die Bedeutung und Notwendigkeit von schriftlichen Festlegungen wurde allen klar.

ProVista® method kit gibt den Studierenden einen Leitfaden durch die einzelnen Aktivitäten und die Rollenverteilung während der Softwareentwicklung. In der abschließenden Bewertung des Projektes wurde von den Studierenden einhellig festgestellt, dass der Verzicht auf ein Konfigurationsmanagement eine Fehlentscheidung war. Auch bei relativ kleinen Projekten ist die Aufgabe des Konfigurationsmanagers nicht zu vernachlässigen.

Zusammenfassend lässt sich feststellen, dass beide Seiten – Lehrende und Lernende – das ProVista® method kit sehr positiv bewerteten.

Einige Teile aus dem Werkzeug werden inzwischen auch in der Vorlesung Softwaretechnik verwendet, um die Praxisrelevanz zu verdeutlichen.

Literatur und Hinweise im Netz (Stand 9.2000)

1. http://www.3soft.de/d/provista/index.php3?select=provista
2. http://www.3soft.de/
3. Oesterreich (Hrsg.), B.; Hruschka, P.; Josuttis, N.; Kocher, H.; Krasemann, H.; Reinhold, M.: Erfolgreich mit Objektorientierung. Oldenbourg, München Wien, 1999
4. http://www.weblearn.hs-bremen.de/

Projektdaten

Das Projekt eUMeL - editor for Unified Modeling Language - wurde an der Hochschule Bremen im Fachbereich Elektrotechnik und Informatik im SS99 und SS00 durchgeführt. Projektbeteiligte waren: Bayram Alpamar, Frank Breidbach, Norman Helm, Wolfgang Meyer, Swen Moczarski, Henning Plump, Sven Riemann, Wiebke Schröder, Enno Siebels und Holger Viehoefer. Betreut wurde das Projekt von Prof. Dr. Andreas Spillner und Prof. Dr. Ulrich Breymann (im SS99).

Copyrightangaben