

Organisation und Projektmanagement der IV

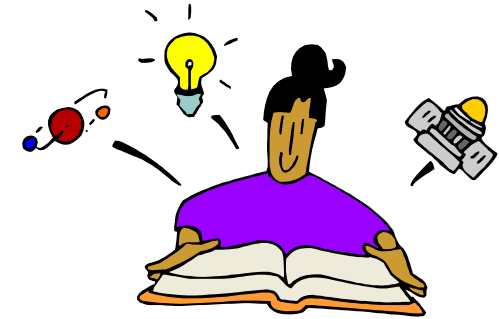
Fallbeispiel Microsoft; Stand: 2002-09-13

Prof. Dr. Dirk Stelzer

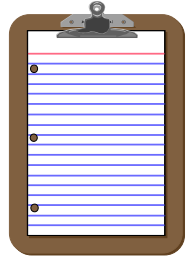


Technische Universität Ilmenau
Fakultät für Wirtschaftswissenschaften
Fachgebiet Informationsmanagement

Lernziele



- Sie können die bisher gelernten Inhalte an praktischen Beispielen anwenden.



Gliederung

- Arbeitsteilung
- Gestaltungsziele der Softwareentwicklung
- Gestaltungsprinzipien der Softwareentwicklung
 - Balance flexibility and structure
 - Synch-and-stabilize
 - Fix resources, evolve features
 - Flexibly changing specs as the product evolves

Empirische Studie 1993-1995

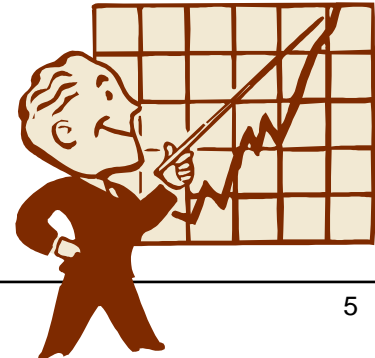


- Durchgeführt von Michael A. **Cusumano**
(Professor für Strategy and Technology Management am MIT, Cambridge, Mass.) und Richard W. **Selby**
(Ass. Professor of Computer Science an der Univ. of California, Irvine)
 - Analyse von Projektdokumenten
 - Interviews mit 38 Führungskräften bei Microsoft

Einige Kennzahlen des Unternehmens Stand: 1996

- 20.500 Mitarbeiter (Durchschnittsalter: ca. 30 Jahre)
- 250 verschiedene Produkte
- 80-85 % Marktanteil bei PC-Betriebssystemen
(z.B. Windows 3.1 weltweit 70 Mio. Installationen)
- 25 % aller weltweit verkauften PC-Standardanwendungssoftware
- 8,7 Mrd. US\$ Umsatz
- Jährliches Umsatzwachstum 1986-1996: [24, 75 %]
- Operating Profit im Durchschnitt 1986-1996 > 30% vom Umsatz

Quelle: Michael A. Cusumano, Richard W. Selby: How Microsoft Builds Software.
In: Communications of the ACM. Nr. 6, 1997, S. 54



Rollen verschiedener Mitarbeiter

- **Product Management** (Marketing Spezialisten)
 - formulieren „vision statement“
(grobe Zielrichtung für das zu entwickelnde Produkt)
 - priorisieren Benutzeraktivitäten,
welche durch das Produkt unterstützt werden sollen
- **Program Management**
 - verantwortlich für funktionale Produktspezifikation
 - verantwortlich für Koordination und Kommunikation
der verschiedenen Entwicklungsteams und zwischen Marketing und Entwicklung
- **Entwickler**
 - entwickeln Produktspezifikation gemeinsam mit Program Management
 - entwickeln „features“ des Produktes selbständig auf der Basis der evolvierenden Produktspezifikation
 - enge Zusammenarbeit mit Program management und Testern
- **Tester**



Aufgaben der Entwickler

- Determine the vision for new features
- Design the features
- Allocate project resources
- Build the features
- Test the features
- Prepare the product for shipping



Zusammenarbeit der Entwickler in „feature teams“

- Produkte haben in der Regel eine funktional-modulare Architektur
- Projektmanager gliedern das zu entwickelnde Produkt in verschiedene „features“
- Jedes „feature“ wird durch ein „feature team“ von 3-8 Mitarbeitern realisiert. (Projektstruktur entspricht der Produktstruktur)
- Jedes feature team entwickelt, integriert, testet und behebt Fehler.
- Sobald ein feature realisiert ist, übernimmt das Team das nächste feature.



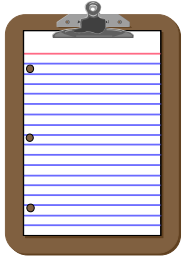
Mitarbeiterstruktur von Microsoft (1995)

- gegliedert nach Aufgabengebieten -

Number	in %	Area or Function
400	2%	Program managers and product planners
1.850	10%	Software design engineers
1.850	10%	Software test engineers
2.100	12%	Customer support engineers
4.000	22%	Marketing, sales, and consulting services
600	3%	User education
2.200	12%	Operations and administration
300	2%	Research
4.500	25%	Overseas personnel (various functions, including 400 developers)
17.800	100%	Total

} total development staff = 4.100

(Quelle: Cusumano, Selby / Microsoft Secrets / 51)



Gliederung

- Arbeitsteilung
- **Gestaltungsziele der Softwareentwicklung**
- Gestaltungsprinzipien der Softwareentwicklung
 - Balance flexibility and structure
 - Synch-and-stabilize
 - Fix resources, evolve features
 - Flexibly changing specs as the product evolves

Gestaltungsziele

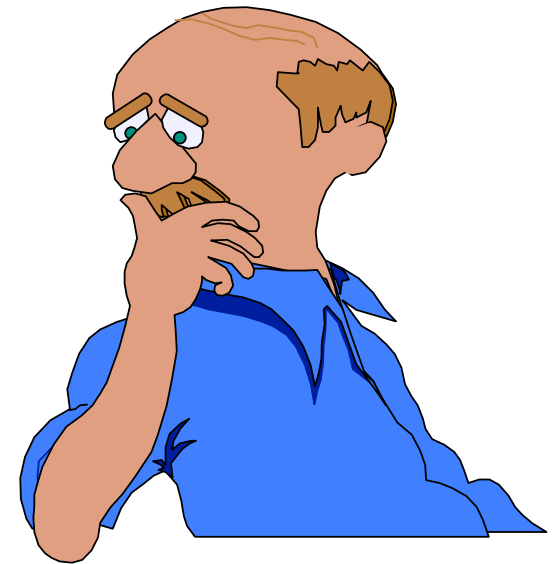
Ziele der Gestaltung der Softwareentwicklung

- Produktqualität
 - (bei Anwendungssoftware)
Die Kundenbedürfnisse, für die eine möglichst große Zahl von Kunden zu zahlen bereit sind
 - (bei systemnaher Software)
Stability, performance, reliability
- Time to market
- Flexibilität / Anpassungsfähigkeit / Innovationsfähigkeit
- „Roll-Out-Fähigkeit“ / „ship-readiness“



„Microsoft's philosophy for product development ...

has been to cultivate its roots as a highly flexible, entrepreneurial company and **not to adopt too many of the structured software-engineering practices** commonly promoted by such organizations as the Software Engineering Institute and the International Standards Organization.“



Michael A. Cusumano, Richard W. Selby: How Microsoft Builds Software.
In: Communications of the ACM. Nr. 6, 1997, S. 54

Gestaltungsprinzipien

- Balance flexibility and structure
- Synch-and-stabilize
- Fix resources, evolve features
- Flexibly changing specs as the product evolves





Balance flexibility and structure

„Microsoft has tried to „scale up“ a **loosely structured small-team** (some might say hacker) style of **product development**.

The objective is to get many small **parallel teams** (three to eight developers each) or individual programmers to work together as a single relatively large team in order to build large products **relatively quickly** while still allowing individual programmers and teams **freedom to evolve** their designs and operate nearly **autonomously**.

These small parallel teams evolve features and whole products **incrementally** while occasionally introducing new concepts and technologies.

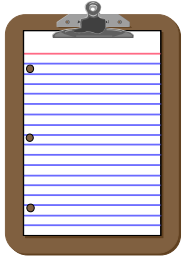
However, because developers are free to innovate as they go along, they must **synchronize** their changes frequently so product components all work together.“

Michael A. Cusumano, Richard W. Selby: How Microsoft Builds Software.
In: Communications of the ACM. Nr. 6, 1997, S. 54

Einige Besonderheiten der Entwicklung

- In der Regel werden keine Designdokumente angefertigt.
- Der Sourcecode ist das einzige (Zwischen-)produkt.
- Entwicklungsarbeit wird in vielen Teams parallel durchgeführt.
- Teams synchronisieren und „debuggen“ ihre Arbeit täglich.
- Es wird angestrebt, für alle relevanten Plattformen und Märkte jederzeit ein auslieferbares Produkt zu haben.
- Kontinuierlicher Test der einzelnen Features und des gesamten Produkts.





Gliederung

- Arbeitsteilung
- Gestaltungsziele der Softwareentwicklung
- Gestaltungsprinzipien der Softwareentwicklung
 - Balance flexibility and structure
 - Synch-and-stabilize
 - Fix resources, evolve features
 - Flexibly changing specs as the product evolves

Synch-and-stabilize

(Microsoft's style of product development)



„Continually **synchronize** what people are doing as individuals and as members of parallel teams, and periodically **stabilize** the product in increments, rather than once at the end of the project.“

- **Daily build**

the act of putting together partially completed or finished pieces of a software product during the development process to see what functions work and what problems exist, usually by completely recompiling the source code and executing automated regression tests.

Michael A. Cusumano, Richard W. Selby: How Microsoft Builds Software.
In: Communications of the ACM. Nr. 6, 1997, S. 54

Daily build (1/2)



- Neu entwickelte oder modifizierte Codekomponenten müssen zu einer bestimmten Uhrzeit „eingescheckt“ sein.
- Im Anschluss werden alle Komponenten zum Master zusammengefügt, kompiliert und (teilweise) automatisch getestet.
- Fehlerhafte Komponenten, welche die vollständige Kompilierung verhindern, müssen sofort korrigiert werden.

Daily build (2/2)



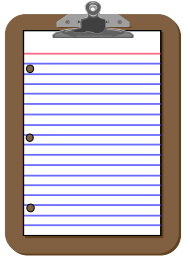
- Developers can check out private copies of source code files from a centralized master version of the source code.
- They implement their features by making changes to their private copies.
- The developers then create a private build of the product containing the new feature and test it.
- They then check in the changes to the master version of the source code.
- The check-in process includes an automated regression test.
- Project build master generates a complete build of the product on a daily basis.
- Daily builds are generated for each platform and for each market.

Sequential Development

- Separate phases done in sequence
- Complete „frozen“ specification and detailed design before building the product
- Trying to build all pieces of a product simultaneously
- One late and large integration and system test at the project's end
- Aiming for feature and product „perfection“ in each project cycle
- Feedback primarily after development as inputs for future projects
- Working primarily as a large group of individuals in a separate functional dep.

Synch-and-Stabilize

- Product development and testing done in parallel
- Vision statement and evolving specification
- Features prioritized and built in 3 or 4 milestone subprojects
- Frequent synchronizations (daily builds) and intermediate stabilizations (milestones)
- „fixed“ release and ship dates and multiple release cycles
- Customer feedback continuous in the development process
- Product and process design so large teams work like small teams

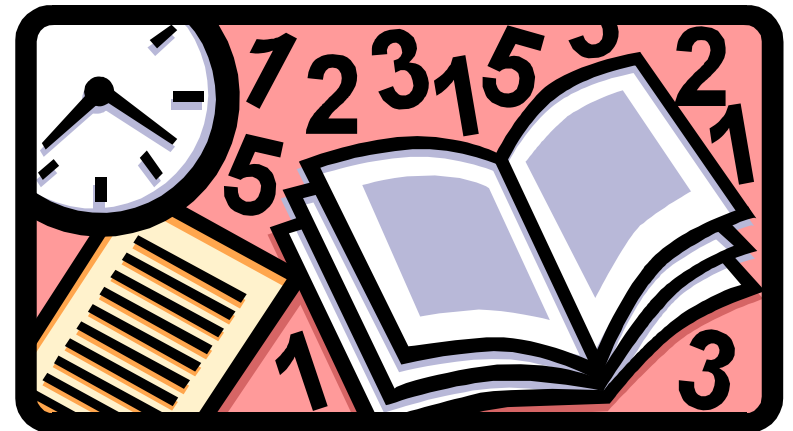


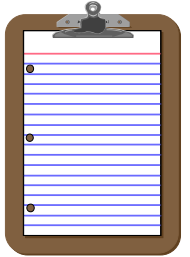
Gliederung

- Arbeitsteilung
- Gestaltungsziele der Softwareentwicklung
- Gestaltungsprinzipien der Softwareentwicklung
 - Balance flexibility and structure
 - Synch-and-stabilize
 - **Fix resources, evolve features**
 - Flexibly changing specs as the product evolves

Fix resources, evolve features

- Entwickler führen Aufwandsschätzungen für Arbeitseinheiten von 0,5 – 3 Tagen durch.
- Entwickler müssen sich verpflichten, diese Pläne einzuhalten.
- Zeitpläne dominieren in der Regel die Funktionalität.
- 20 bis 30 % der Specifications verändern sich im Durchschnitt im Entwicklungsverlauf.
- Weitere 20 bis 25 % der ursprünglichen Specifications werden gestrichen





Gliederung

- Arbeitsteilung
- Gestaltungsziele der Softwareentwicklung
- Gestaltungsprinzipien der Softwareentwicklung
 - Balance flexibility and structure
 - Synch-and-stabilize
 - Fix resources, evolve features
 - Flexibly changing specs as the product evolves

Flexibly changing specs as the product evolves

- Product Management formuliert das **Vision statement**, welches die Zielsetzung des Produktes beschreibt
- Program manager formulieren die **feature specifications**, als Richtlinie für die Entwickler
- Entwickler detaillieren die specifications während der Entwicklung
- Specifications
 - haben Richtliniencharakter
 - sind eher Output als Input der Entwicklung
 - beschreiben eher, was das Produkt nicht tun soll, als das, was es tun soll
 - werden (durch Beobachtung potentieller Kunden) kontinuierlich aktualisiert

Kontroll- und Verständnisfragen



- Welche Form der Arbeitsteilung ist bei Microsoft vorherrschend?
Begründen Sie Ihre Antwort.
- Welche Koordinationsmechanismen werden in der Softwareentwicklung bei Microsoft vorwiegend verwendet?
Begründen Sie Ihre Antwort.



Literaturhinweise

- Michael A. Cusumano, Richard W. Selby: Microsoft Secrets: How the world's most powerful software company creates technology, shapes markets, and manages people. New York 1995.
- Michael A. Cusumano, Richard W. Selby: Die Microsoft Methode. Sieben Prinzipien, wie man ein Unternehmen an die Weltspitze bringt. Freiburg 1996.
- Michael A. Cusumano, Richard W. Selby: How Microsoft Builds Software. In: Communications of the ACM. Nr. 6, 1997, S. 53-61.