Dinosaur meets Archaeopteryx?

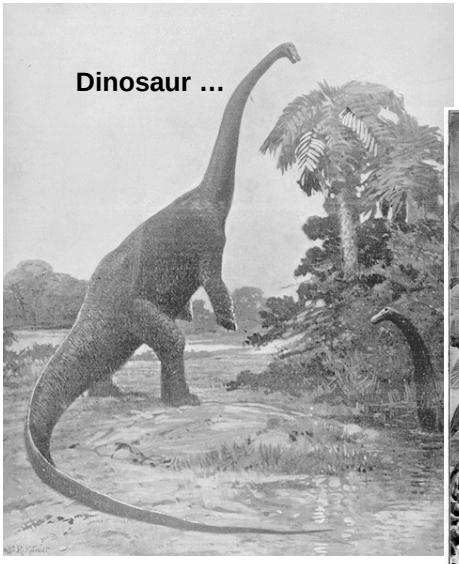
Seven Theses on Rational's Unified Process (RUP)

Wolfgang Hesse

c/o FB Mathematik und Informatik, Universität Marburg,
Hans Meerwein-Str., D-35032 Marburg

email: hesse@informatik.uni-marburg.de

Dinosaur ...



meets ...

Archaeopteryx (?)

Seven theses on the RUP

Thesis 1: The RUP is still too much "waterfall"-like.

Thesis 2: The RUP is not architecture-centric enough

Thesis 3: RUP iterations should be attached to building blocks - rather than to phases

Thesis 4: RUP workflows (now: disciplines) are unnecessarily complex. The (formerly) so-called "core workflows" are just activity categories.

Thesis 5: The RUP ignores most powerful mechanisms for mastering complexity such as hierarchy, recursion and orthogonality

Thesis 6: The RUP does not appropriately support management. Its milestone concept is too weak.

Thesis 7: The RUP does not satisfactorily address the various groups and roles in the software process, in particular the user role.

Claims and core elements of the RUP

Claims

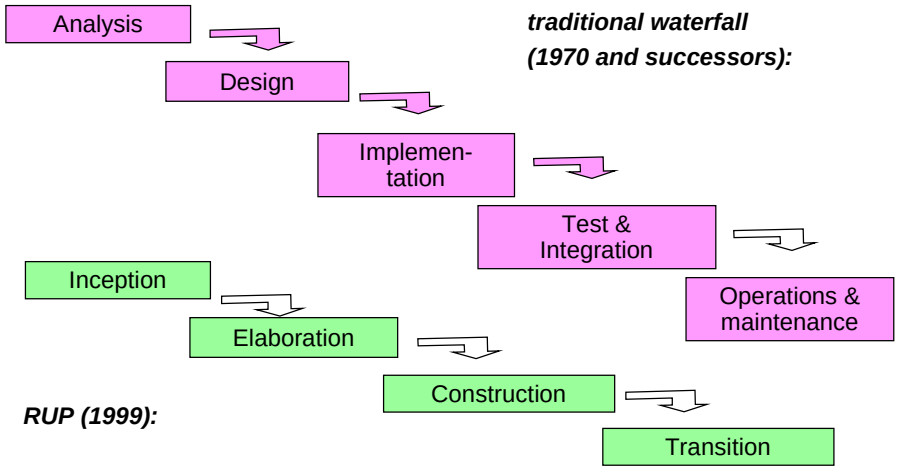
- *use case driven*
- *architecture centric*
- *iterative and incremental*

Core elements:

- Phases
- Iterations
- Core workflows

RUP-DmA 50

Thesis 1: ***The RUP is based on a phase oriented software life cycle model which is no longer adequate to support most contemporary development approaches.***



RUP-DmA 60

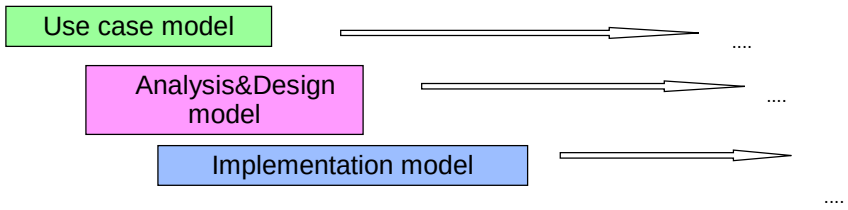
Do phases offer an adequate process structure?

- Benefits and problems of ***phases*** have been debated for long time.
- Contemporary software processes are highly ***complex***, ***differentiated*** and ***multi-faceted***.
- They consist of many, heterogeneous ***sub-processes*** typically running in parallel. Synchronisation of sub-processes should not be not phase-controlled but ***demand-controlled***.
- Phases offer just a superficial, rough and ***uppermost-level*** structure.
- Complex systems are organised in hierarchies and self-resembling sub-structures. Why don't we organise the corresponding processes in an analogous way?

What is needed now - is not new names for old phases, but other, ***more differentiated, systematic and appropriate*** process structures and synchronisation mechanisms.

Thesis 2: ***In contrast to its authors' claims, RUP is not an "architecture centric" process but it is still dominated by phase structure.***

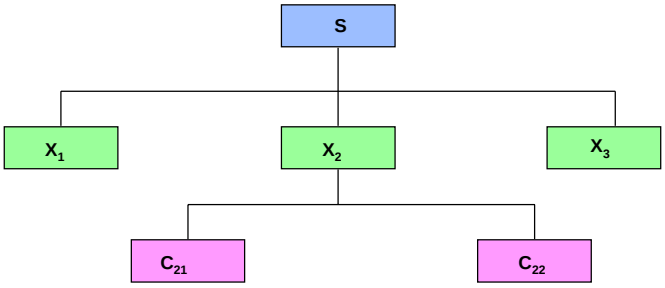
RUP "architecture":



If we take the OO paradigm serious ...

.. the three "models" should merely embody three ***states*** of ***one and the same*** model.

A simple but powerful architecture



S: System
X_i : Components
C_{ij} : Classes

- **Transaction-oriented process** (RUP et al.)

  Activities aim at refining the models.

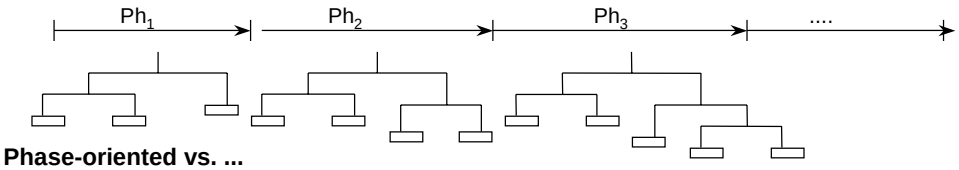
Alternative:

- **Document-oriented process** (Ref.: Denert [Den 93]):

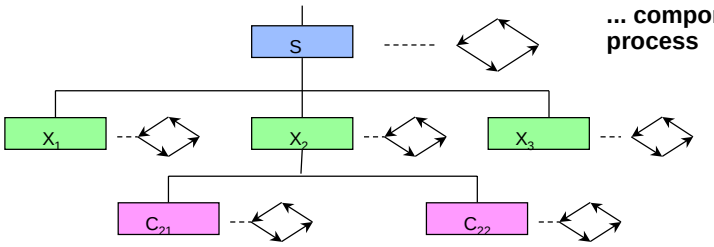
  Activities lead to defined results (*documents, building blocks)* which are developed (relatively) independent from each other and then integrated.

RUP-DmA 90

Thesis 3: ***The RUP does well in introducing iterations in the software development process, but there is much less need for phase iterations than for (sub-) product development cycles.***



Phase-oriented vs. ...



... component-oriented process

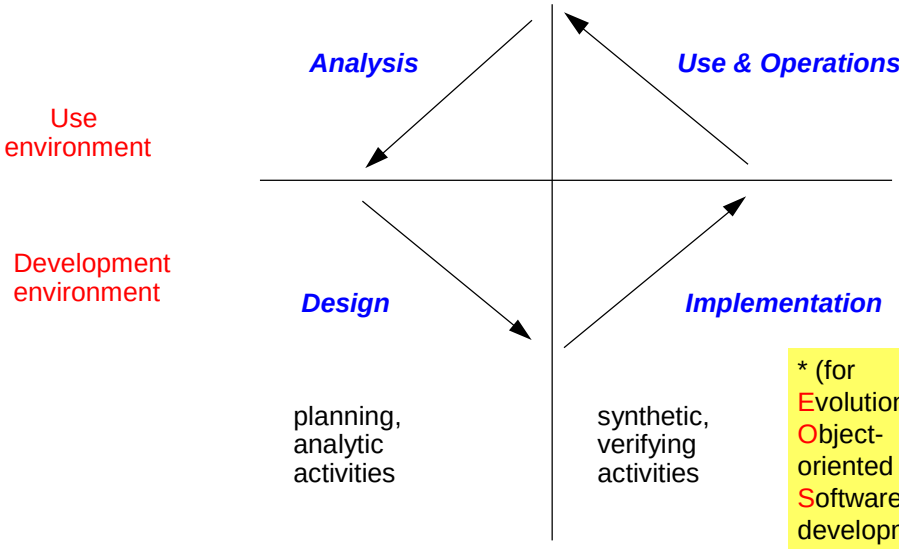
Legend:

Building block

Phase or activity

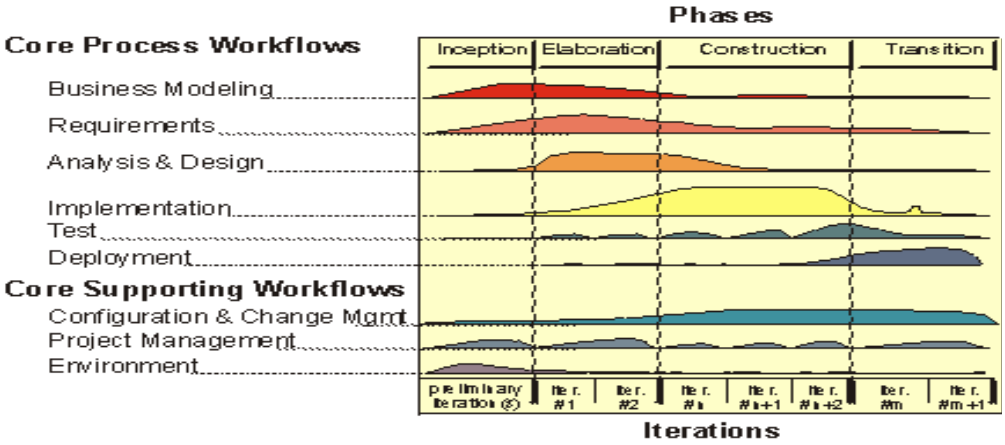
RUP-DmA 100

Activities of a development cycle in the EOS* model



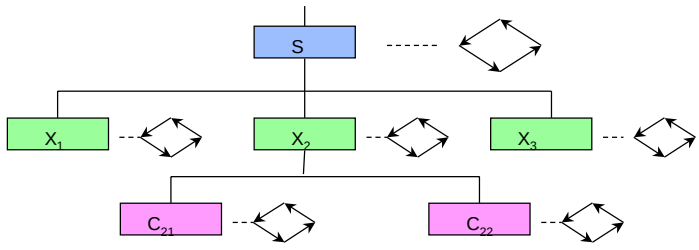
* (for Evolutionary, Object-oriented Software development)

Thesis 4: ***The RUP concept of workflows - now called "disciplines" - adds unnecessary complexity to the process.***



The former "core workflows" are misnamed and just activities of the same or a similar kind. They overlap with phases in a confusing way and do not contribute to a clear, transparent process structure.

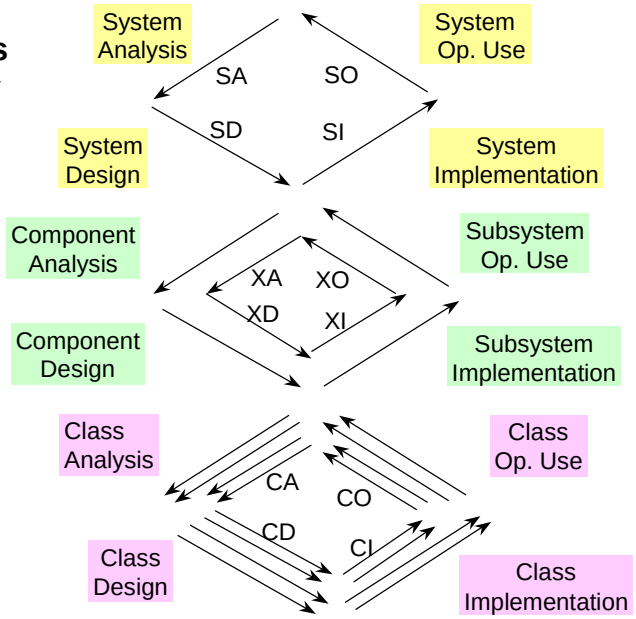
Thesis 5: ***The RUP does not offer appropriate support for structuring complex software processes. It ignores most powerful mechanisms of computer science for mastering complexity: hierarchy, recursion and orthogonality.***



Orthogonal system & process structure

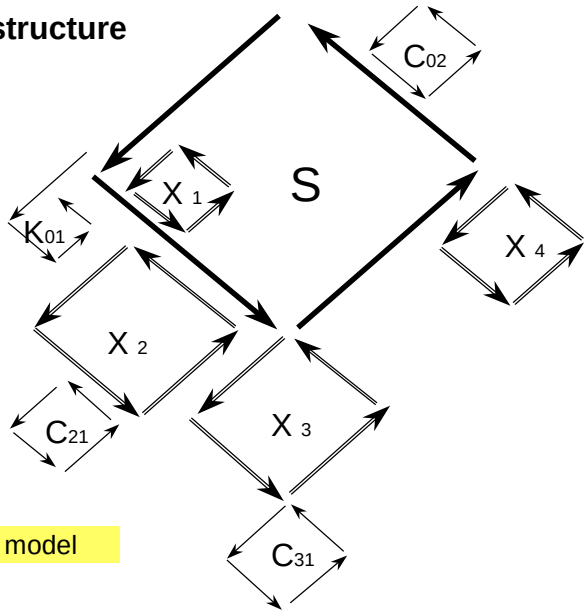
RUP-DmA 130

Combining development cycles in a traditional way



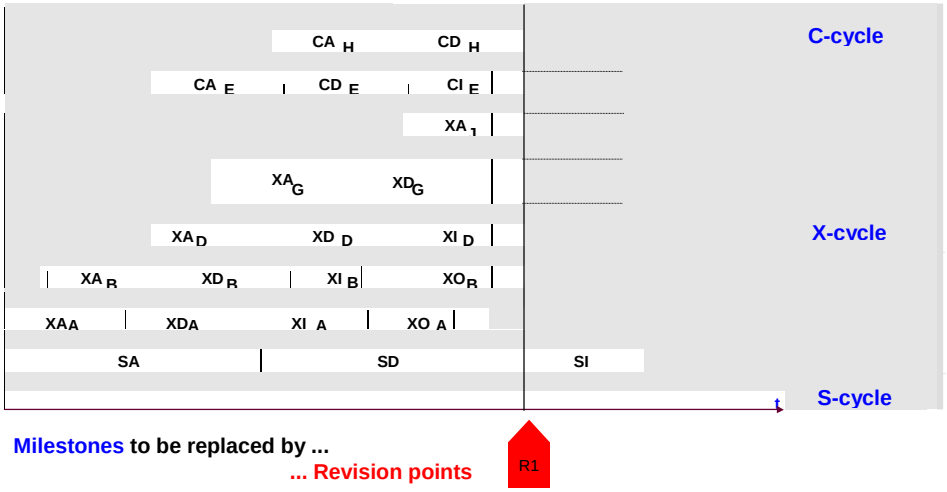
RUP-DmA 140

A "fractal" process structure



.. as proposed in the EOS model

Thesis 6: ***Due to its lack of transparency and structural flexibility, the RUP does not appropriately support management - in particular that of large projects. The RUP concept of milestones is too weak for complex coordination tasks.***

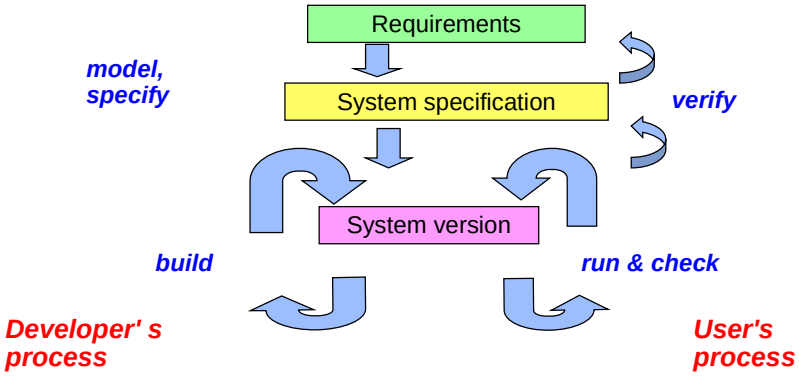


Milestones to be replaced by ...

... Revision points

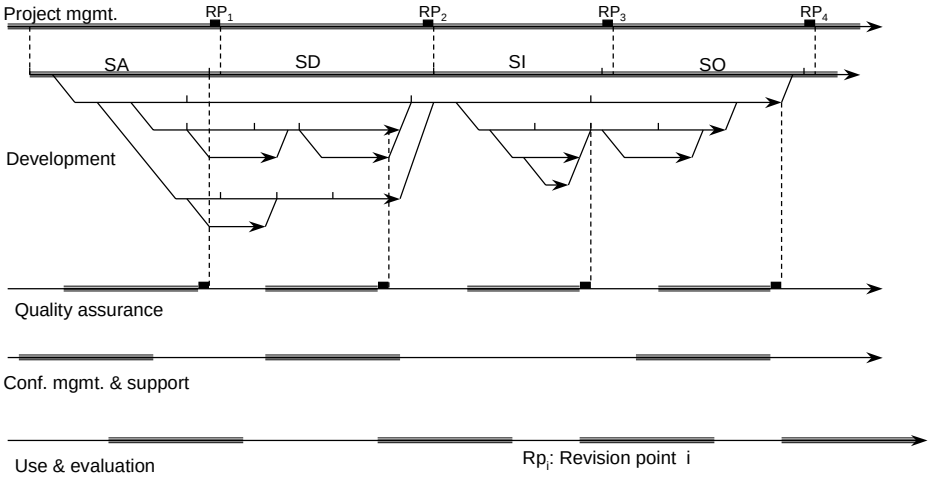
Thesis 7: ***The RUP does not satisfactorily address the roles and interactions of various groups concerned with the software process, in particular the role of the users and their feedback on the process is neglected.***

Cf. the STEPS model (Floyd et al. 1989): Co-operation of developers and users



RUP-DmA 170

A general process model



SA: System analysis
SD: System design
SI: System implementation
SO: System operations & use

Reference: [Hes 97b]

RUP-DmA 180

Summary and outlook

RUP: is principally a respectable approach to bring order into the jungle of OO process models

But:

- Approach is too traditional
- RUP is transaction-oriented instead of result-oriented
- Components (and their development) are not given adequate attention
- Use-case driven and incremental development is considered "the only way"
- "Unification" of processes is problematic - in particular when the habits and practices of people are touched

More promising (?):

"Multi-variant approach" based on a "toolbox" of standard activities, result types, roles etc. (cf. [H-N 99])

References

[Boo 94] G. Booch: *Object-Oriented Analysis and Design with Applications;* Second Edition, Benjamin/Cummings Publ. Comp. 1994

[Den 93] E. Denert: Dokumentenorientierte Software-Entwicklung, in: *Informatik-Spektrum 16/3,* pp. 159-164 (1993)

[FRS 89] Ch. Floyd, F.-M. Reisin, G. Schmidt: STEPS to software development with users; in: *C. Ghezzi, J. McDermid (eds.): ESEC '89, Second European Software Engineering Conference.* LNCS 387, pp. 48-64; Springer 1989

[Hes 96] W. Hesse: Theory and practice of the software process - a field study and its implications for project management; in: C. Montangero (Ed.): *Software Process Technology, 5th European Workshop, EWSPT 96, Springer LNCS 1149,* pp. 241-256 (1996)

[Hes 97a] W. Hesse: From WOON to EOS: New development methods require a new software process model; Bericht Nr. 12, Fachbereich Mathematik, Univ. Marburg; and: *Proc. WOON ´96,* St. Petersburg 1997

[Hes 97b] W. Hesse: Improving the software process guided by the EOS model. In: *Proc. SPI '97 European Conference on Software Process Improvement*. Barcelona 1997

[Hes 01a] W. Hesse: RUP - A process model for working with UML - Critical Comments on the Rational Unified Process - Book chapter in: K. Siau et al. (eds): *Unified Modeling Language.* Idea Group Publ. 2001

[Hes 01b] W. Hesse: Dinosaur Meets Archaeopteryx? Seven Theses on Rational's Unified Process (RUP). Proc. CAiSE'01/IFIP 8.1 Int. Workshop on Evaluation of Modeling Methods in System Analysis and Design (EMMSAD'01), Ch. VII, Interlaken 2001

[H-N 99] W. Hesse, J. Noack : A Multi-Variant Approach to Software Process Modelling. In: *M. Jarke, A. Oberweis (Eds.): CAiSE'99*, LNCS 1666, pp. 210-224 (1999)

References (cont'd)

[Jac 93] I. Jacobson: *Object-Oriented Software Engineering - A Use Case Driven Approach;* Revised Printing, Addison- Wesley 1993

[JBR 99] I. Jacobson, G. Booch, J. Rumbaugh: *The Unified Software Development Process.* Addison-Wesley 1999

[Kruc 95] P.B. Kruchten: The 4+1 view model of architecture. *IEEE Software 12 (6),* pp. 42-50 (1995)

[Kruc 99] P.B. Kruchten: *The Rational Unified Process (An Introduction).* Addison Wesley 1999

[Roy 98] W. Royce: *Software Project Management - A Unified Framework,* Addison Wesley 1998

[Rum 91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen*: Object-Oriented Modelling and Design*; Prentice Hall 1991

[Sce 00] K.D. Schewe: UML: A Modern Dinosaur? – A Critical Analysis of the Unified Modelling Language. *Proc. 10th European-Japanese Conf. on Information Modelling and Knowledge Bases.* Saariselkä/Finland

[UML 99] OMG Unified Modelling Language Specification Version 1.3, June 1999. http://www.rational.com/uml/resources/documentation.

Authors address:

Prof. Dr. Wolfgang Hesse

FB Mathematik und Informatik, Universität Marburg,
Hans Meerwein-Str., D-35032 Marburg

Tel.: +49-6421-281515, Fax: +49-6421-285419,
email: hesse@informatik.uni-marburg.de