Formal comparison of methods

Session 6b
25 February 2019

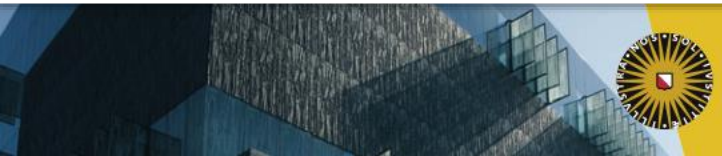
Prof.dr. Sjaak Brinkkemper
Dr. Sietse Overbeek

Universiteit Utrecht

Outline

- We are standing on the shoulders of giants…
  - Method engineering
  - Creation of methods
  - Tool support
  - Method comparison

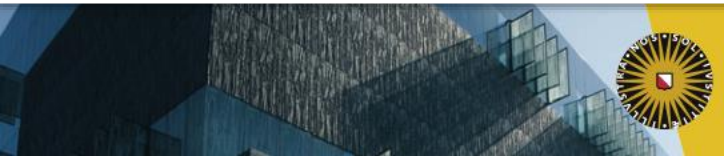
- …but do not always agree with each other



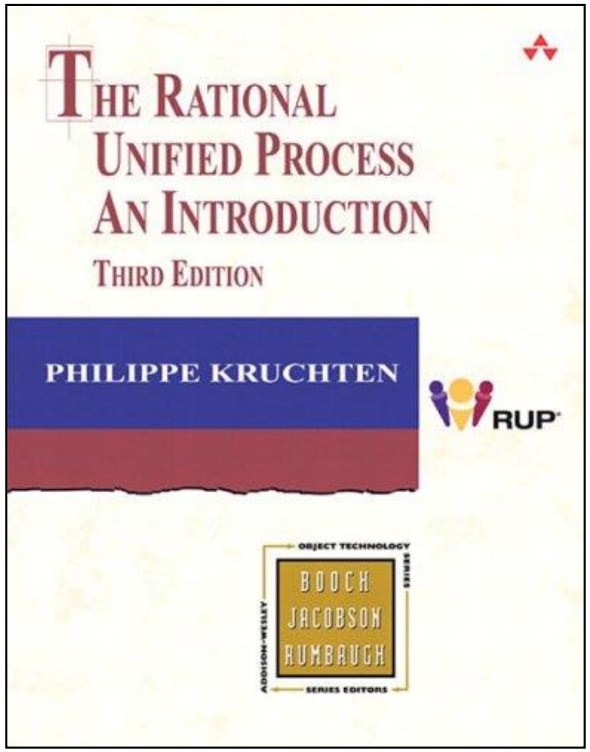
Some questions

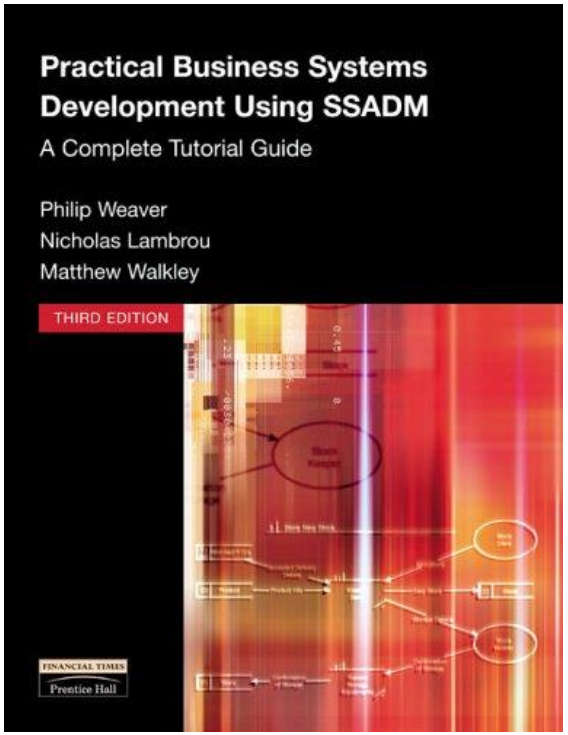
- Which method for the development of information systems is better?
- Should we change our standard method?
- Which technique is better suited for this application?
- What tool should we select to support this method/technique?

- All these questions involve some kind of comparison.
- The objective of this lecture is to provide a theory of an objective method comparison based on method engineering principles.



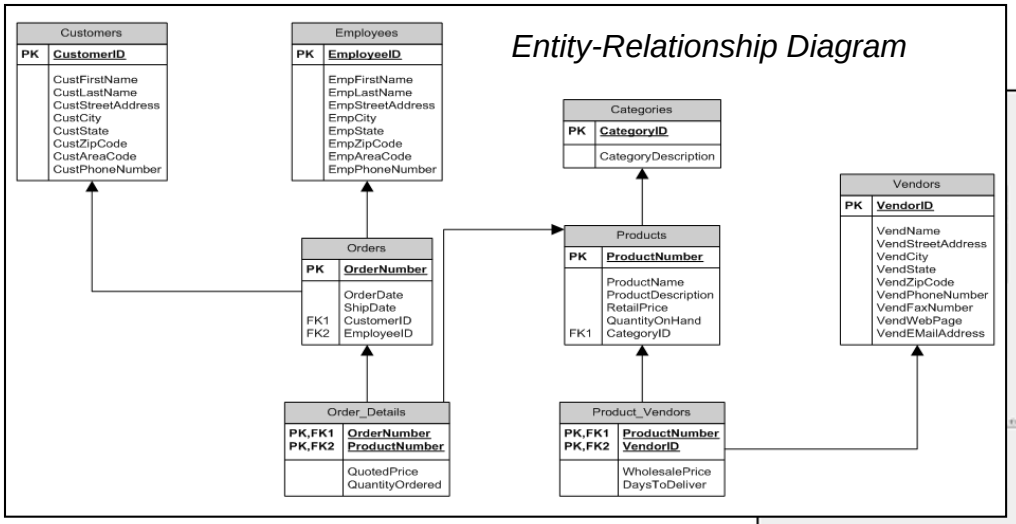
Comparison of methods

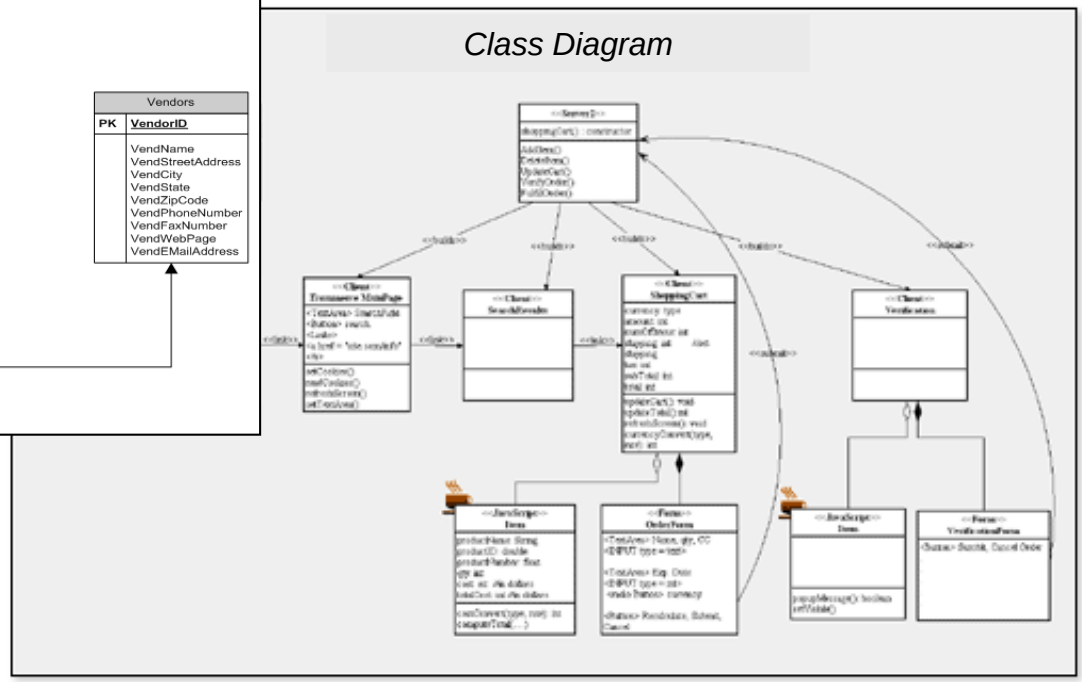




Rational Unified Process versus
Structured Systems Analysis and Design Method

Comparison of techniques

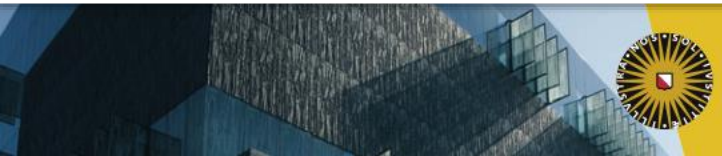




Entity Relationship Diagram versus Class Diagram

Evaluation criteria

- Effectiveness: method adequately suited for the communication with the team and the customer
- Efficiency: method executed within time and budget
- Execution: more deliverable content per time unit
- Correctness: less error repairs per deliverable
- Satisfaction: better acceptance by users and customers
- Learnability: faster or simpler to learn



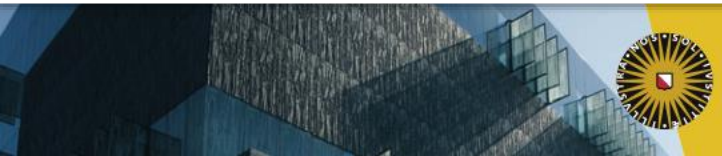
Dilemma of comparison

- How can methods be measured in a scientific valid way?
- This means, the comparison should be independent from:
  - performers
  - domain
  - technology
  - project situation
  - experience
- Arrangement of large field studies or laboratory tests are unfeasible

4-DAT evaluation framework

- 4-DAT (4D Analytical Tool) evaluation framework
- Assessment of 4 degrees of agility for existing or about-to-constructed methods
- Partly qualitative and quantitative
- 4 dimensions:
  - Method scope
  - Agility characterization
  - Agile value characterization
  - Software process characterization

(*Qumer & Henderson-Sellers, 2006)*



DAT-dimensions (excerpt)

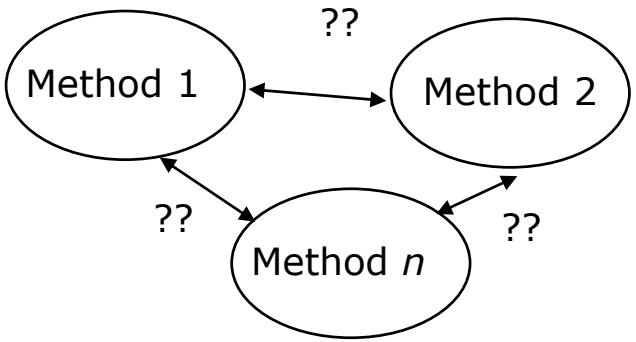
Dimension 1 (Method Scope)

Scope	*Description*
1. Project Size	Does the method specify support for small, medium or large projects (business or other)?
2. Team size	Does the method support for small or large teams (single or multiple teams)?
3. Development style	Which development style (iterative, rapid) does the method cover?
4. Code style	Does the method specify code style (simple or complex)?
5. Technology environment	Which technology environment (tools, compilers) does the method specify?
6. Physical environment	Which physical environment (co-located or distributed) does the method specify?
7. Business culture	What type of business culture (collaborative, cooperative or non-collaborative) does the method specify?
8. Abstraction mechanism	Does the method specify an abstraction mechanism (object-oriented, agent-oriented)?

Dimension 2 (Agility Characterization)

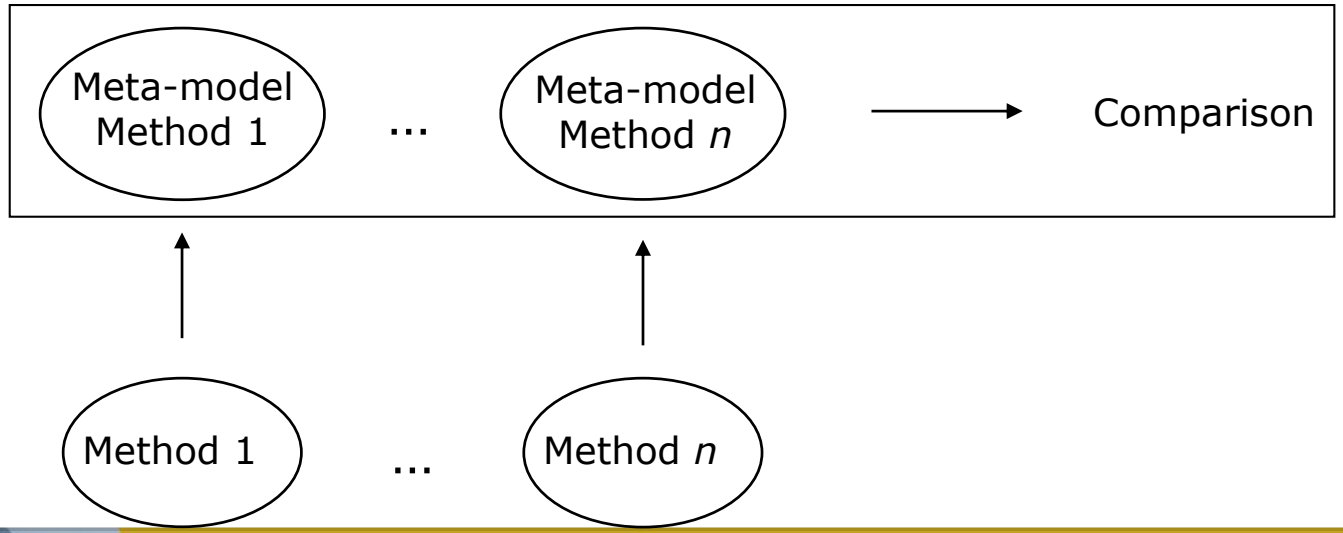
Features	*Description*
1. Flexibility	Does the method accommodate expected or unexpected changes?
2. Speed	Does the method produce results quickly?
3. Leanness	Does the method follow the shortest time span, use economical, simple and quality instruments for production?
4. Learning	Does the method apply updated prior knowledge and experience to create a learning environment?
5. Responsiveness	Does the method exhibit sensitiveness?

Formal comparison



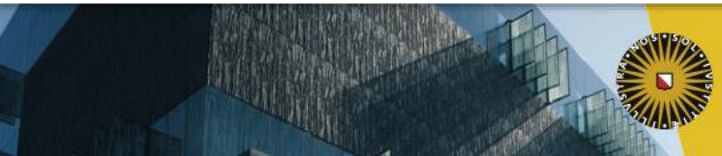
How to compare more than 2 methods?

Arrangement of the comparison

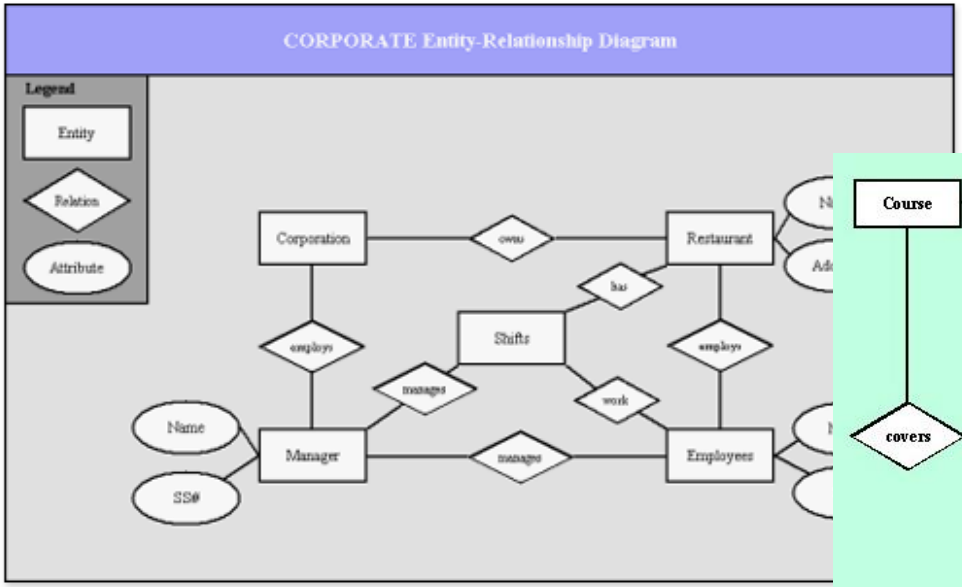


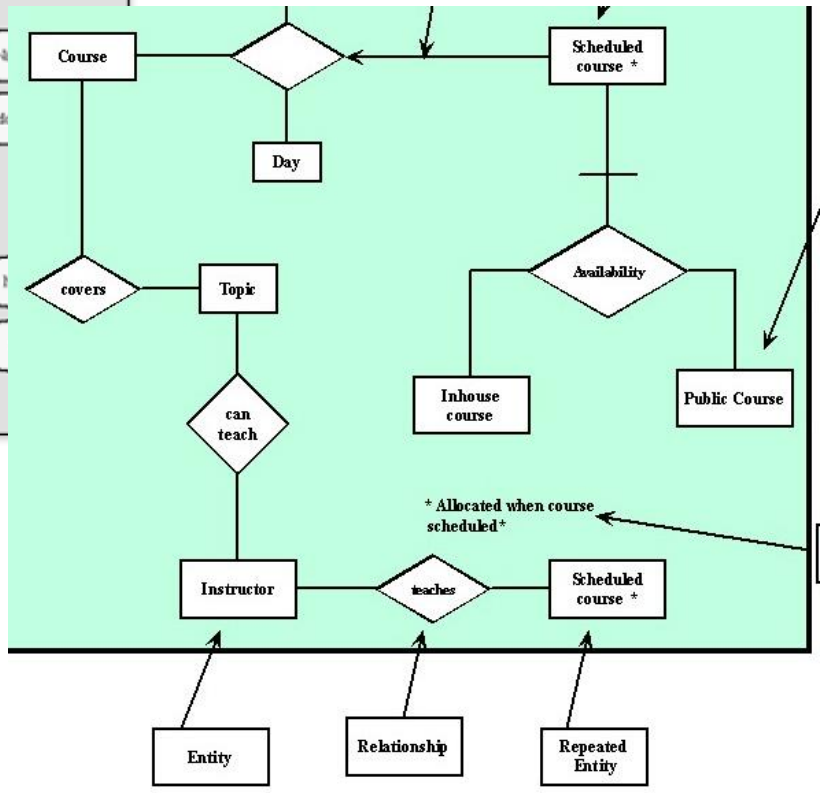
Meta-models

- Usage of meta-models for comparison
- Meta-data models
  - Concepts, associations, properties, ...
- Meta-process models
  - Activities, sub-activities, roles, ...



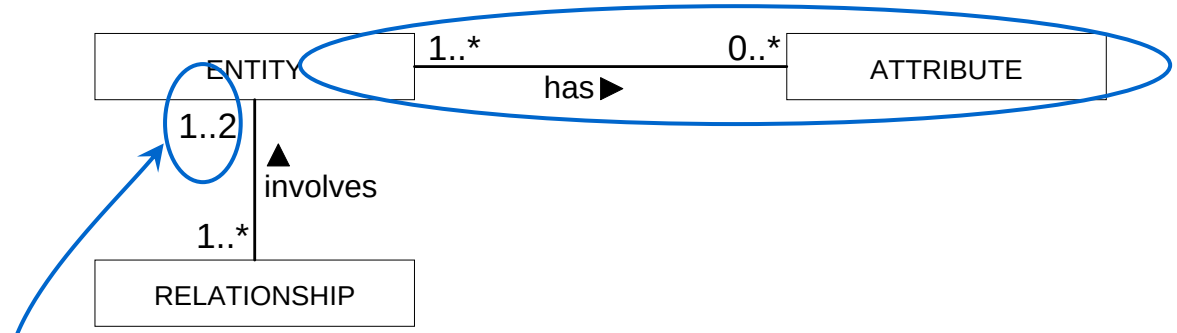
Example (1)


CORPORATE Entity-Relationship Diagram
Legend
Entity
Relation
Attribute
Corporation
owns
Restaurant
has
Shifts
employs
employs
manages
work
Name
Manager
manages
Employees
SS#

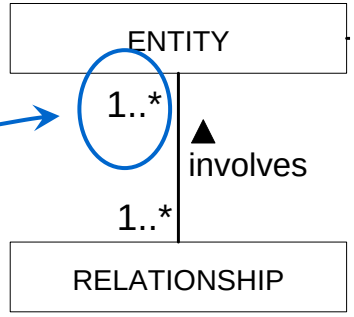

Course
Scheduled course *
Day
covers
Topic
Availability
Inhouse course
Public Course
can teach
* Allocated when course scheduled*
Instructor
teaches
Scheduled course *
Entity
Relationship
Repeated Entity

Example (2)

Binary ERD with attributes



N-ary ERD without attributes

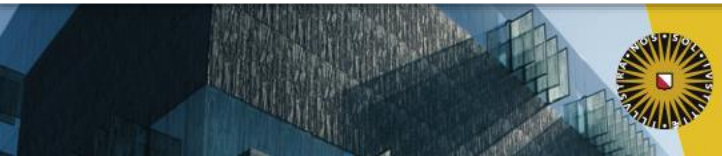


Metrics of meta-models

- Modeling the techniques in OPRR language (Object, Property, Relationship, Role)
- OPRR is the meta-model behind the Meta-Edit tool.

- Analysis of the techniques
  - Count of object types, relationship types and property types
  - Aggregate metrics to measure complexity based on the above counts

Rossi and Brinkkemper (1997)



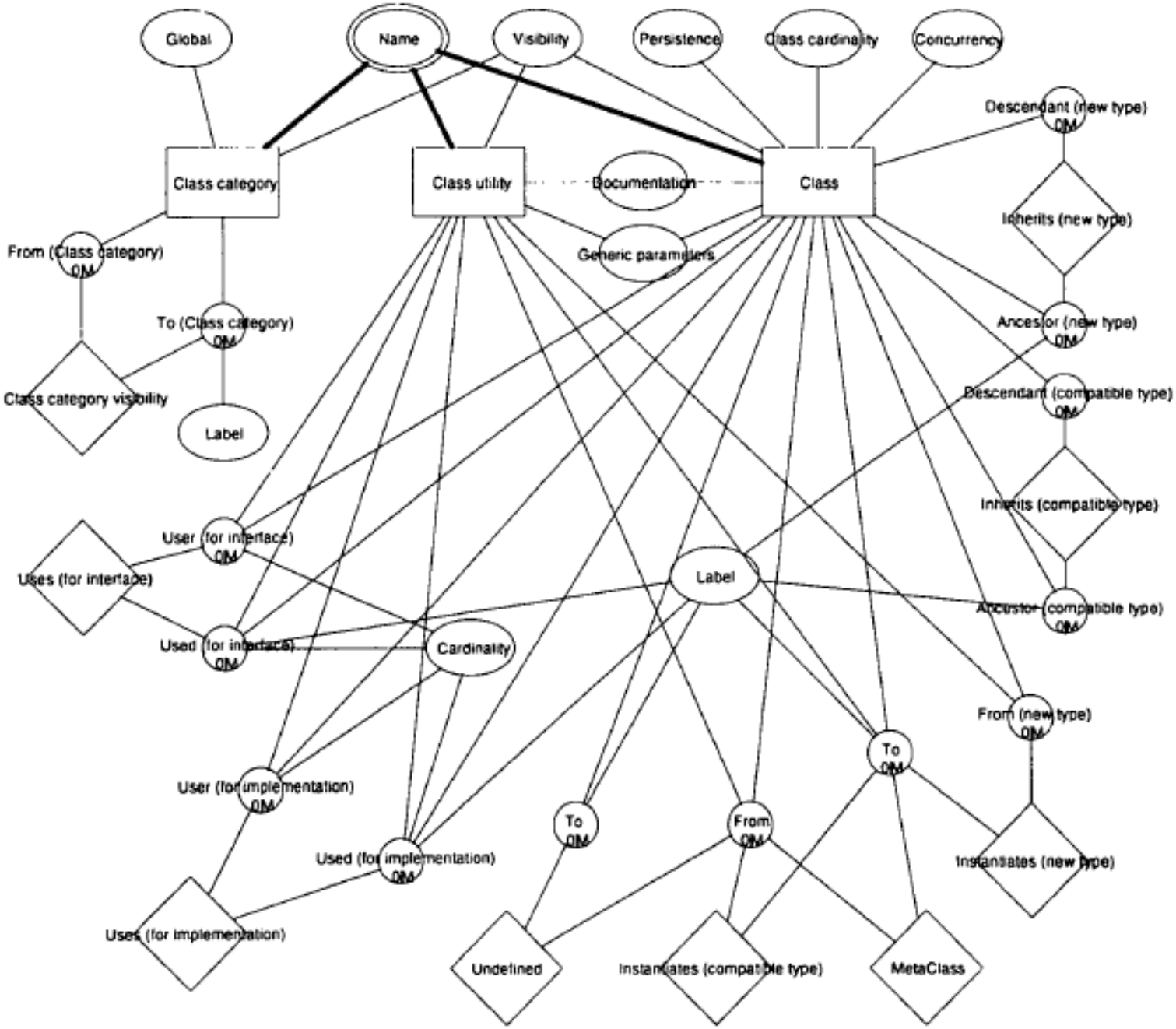


Figure 1. OPRR model of OODA CD

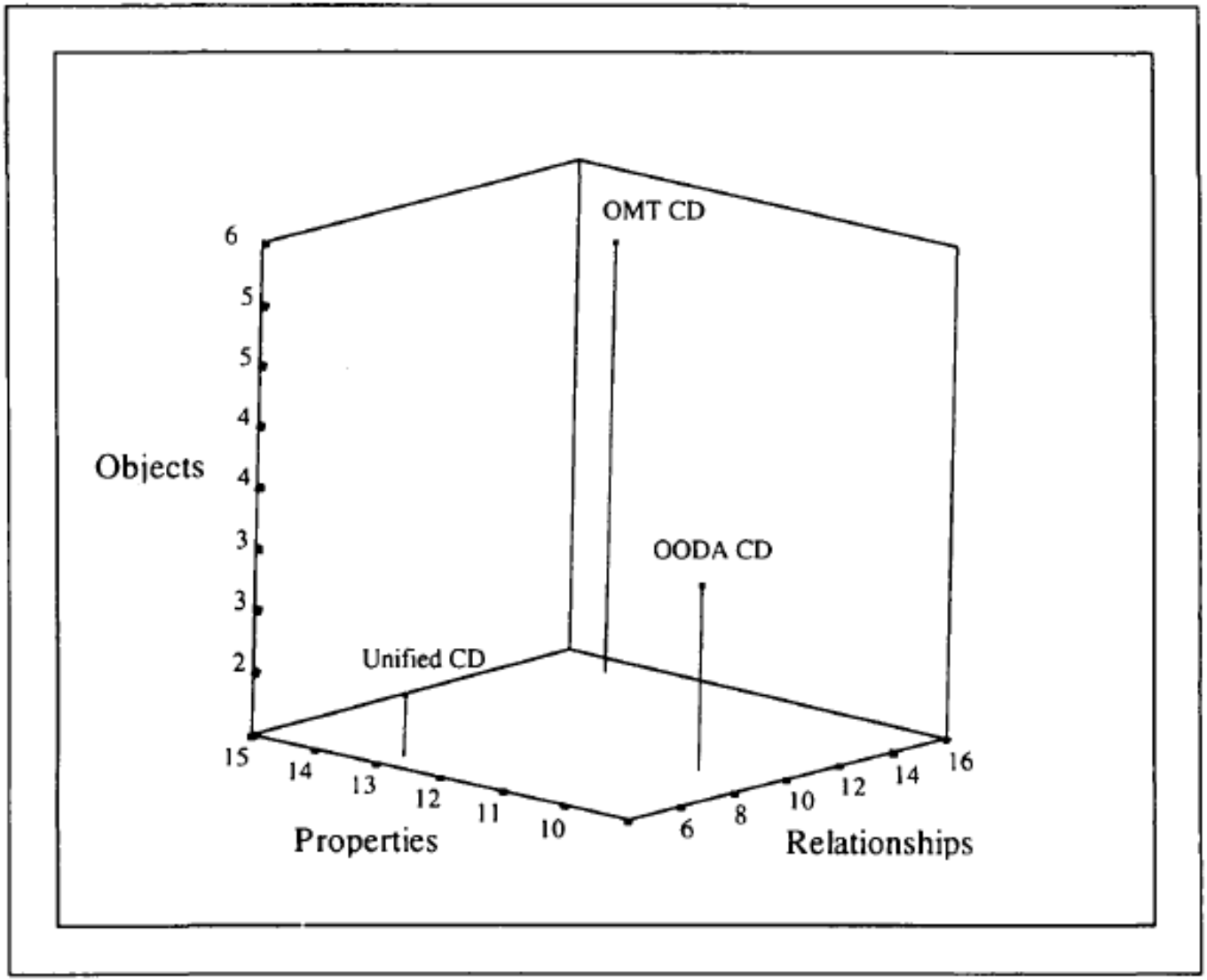


Figure 5. Technique cube

Conceptual differentiation of meta-models

Comparison based on conceptual differentiation of meta-models (*Hong, Goor & Brinkkemper (1993))*

Process:

1. Method selection
2. Method characterization
3. Meta-modeling
4. Creation of a super-method
5. Method comparison

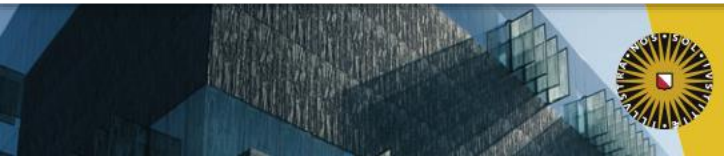
Extensions (*van de Weerd, de Weerd & Brinkkemper (2006))*

Use of the PDD as meta-modeling language

Use of activity and concept tables

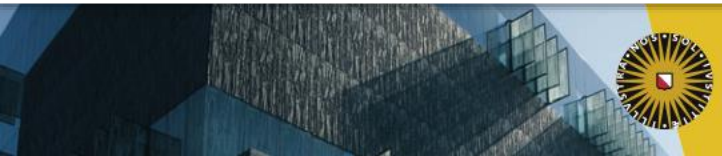
Amanatiadou & van de Weerd (2009)

Comparing methods of companies with similar sitiational factors in order to identify different routes



Execution of the comparison process

1. Method selection
2. Characterization of the chosen methods
3. Meta-modeling
   - Meta-process models
   - Meta-data models
4. Creation of a super-method
5. Method comparison
   - Activities
   - Concepts
   - (Techniques)

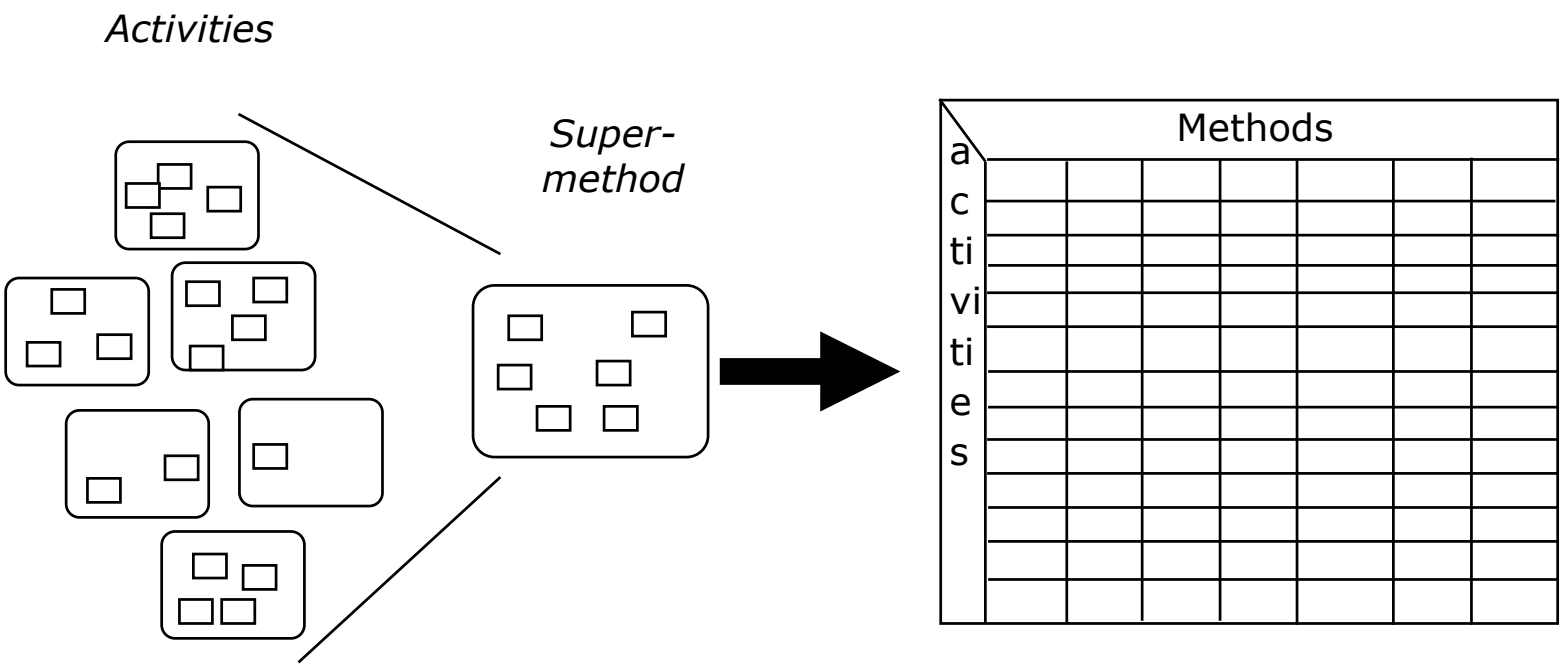


Super-method

- The super-method is a kind of smallest common denominator of the methods in the study,
  i.e.
  a method which contains all activities and concepts that appear in *at least one* of the methods

  Note: in the paper of Hong et al. the super-method is called supermethodology, because it was published in the US

Comparison of activities



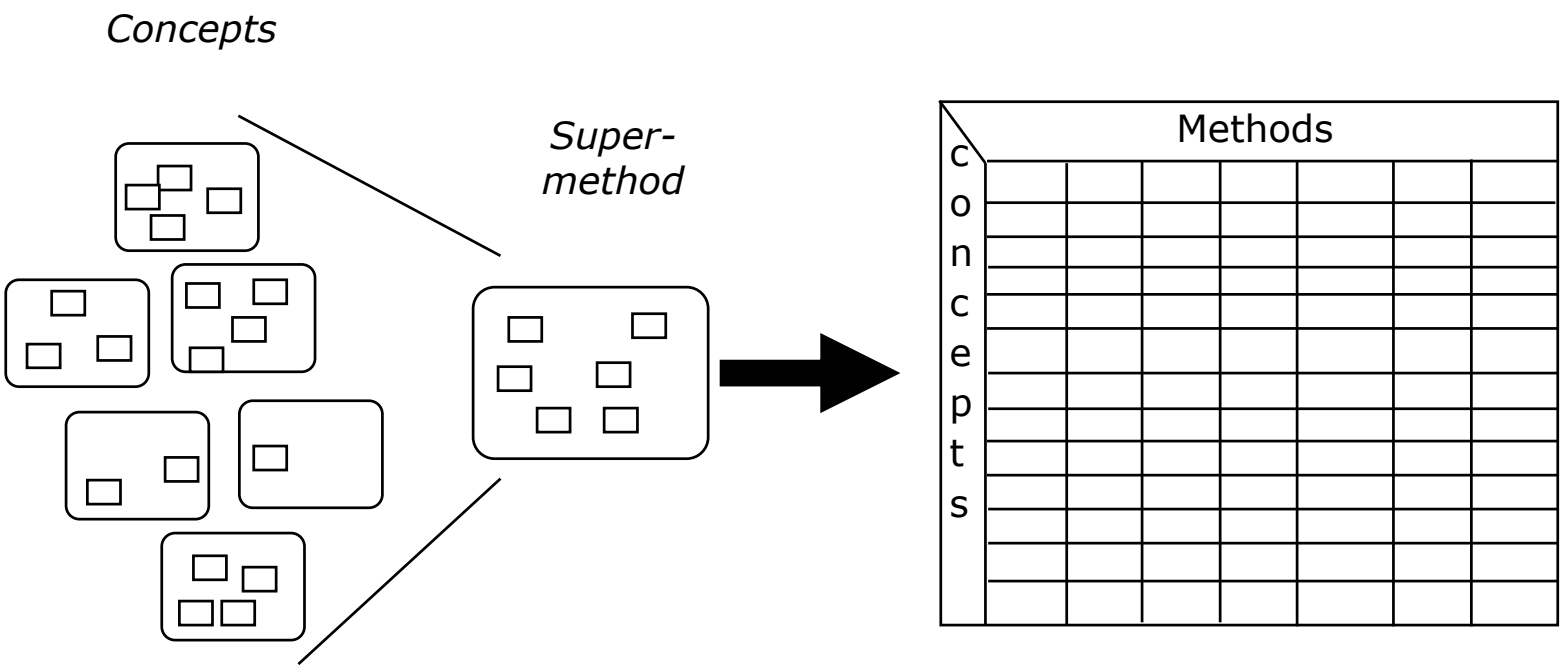
Comparison indicators

s '=' *m*	Activity *s* is equivalent to activity *m*
s '>' *m*	Activity *s* does more than activity *m*
s '<' *m*	Activity *s* does less than activity *m*
s '><' *m*	Activity *s* overlaps partly activity *m*
Blank	Activity *s* is absent from the method

Activity comparison

2. Preproduction Phase	GD&P	ITGD	TGPH	Zylom
2.1 Create game design				
2.1.1 Brainstorm	=			
2.1.2 Delegate design	=			
2.1.3 Write game design document	=	>		=
2.1.4 Evaluate game design document				=
2.1.5 Write technical design document	=	><		<
2.1.6 Create visualizations				=
2.1.7 Present game design				=
2.1.8 Evaluate technology			=	
2.1.9 Define tools and pipeline			=	
2.1.10 Create documentation			=	

Comparison of concepts



Comparison indicators	*s '=' m* *'string'*	Concept *s* is equivalent to concept *m* Concept s is equivalent to concept m but the term *'String'* is used.

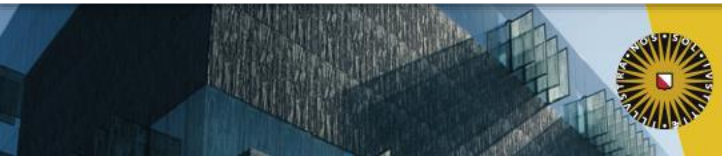
Concept comparison

2. Preproduction Phase	GD&P	ITGD	TGPH	Zylom
2.1 Create game design				
2.1.1 GAME DESIGN DOCUMENT	=	=	DESIGN DOC.	=
2.1.2 CORE GAMEPLAY	=			
2.1.3 CONTEXTUAL GAMEPLAY	=			
2.1.4 STORY	=			
2.1.5 TECHNICAL DESIGN DOCUMENT	=	=	TECHNICAL DOC.	=
2.1.6 REQUIREMENT	ASSET	FEATURE	FEATURE	ASSET
2.1.7 VISIBLE REQUIREMENT	=			
2.1.8 NONVISIBLE REQUIREMENT	=			
2.1.9 FEATURE LIST			=	ASSET LIST
2.1.10 DOCUMENTATION			=	
2.1.112 PROTOTYPE				=

A reference method for game production

- Master's thesis research Stefan de Weerd
- Duration: 7 months (2006/2007)
- Domain: professional game production

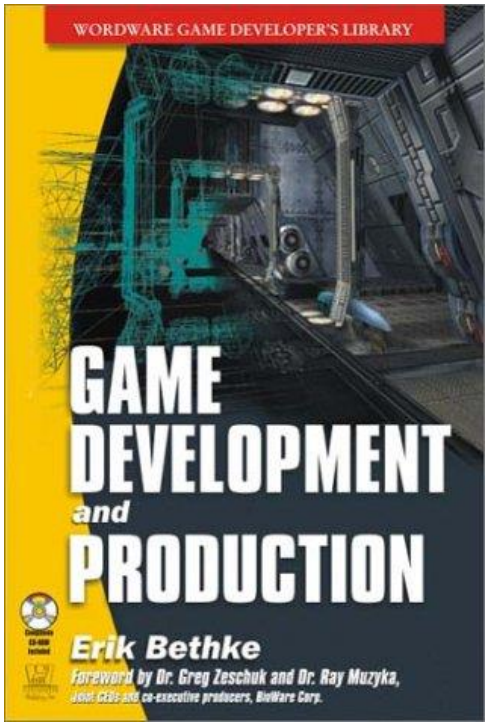
- 3 books for game development
- Zylom case study



Game development and production

Eric Bethke, 2003

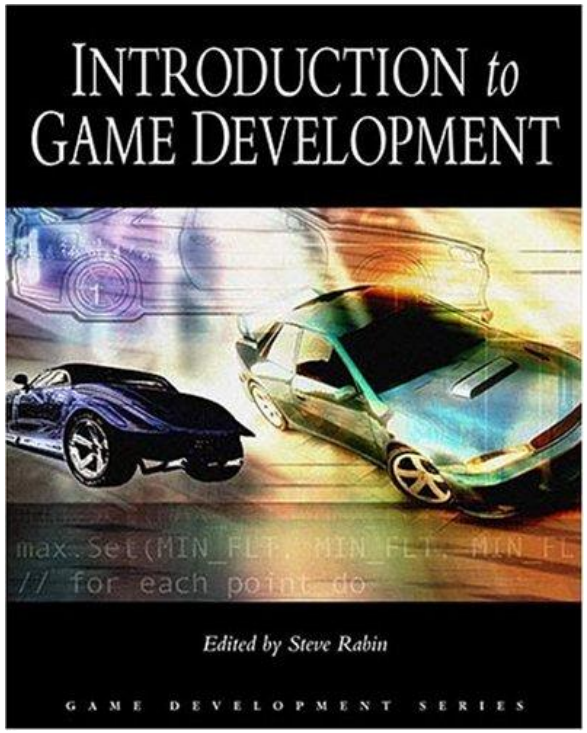
- Focus
  - Deliverables
  - Practical issues
  - Management support



Introduction to game development

Steve Rabin, 2005

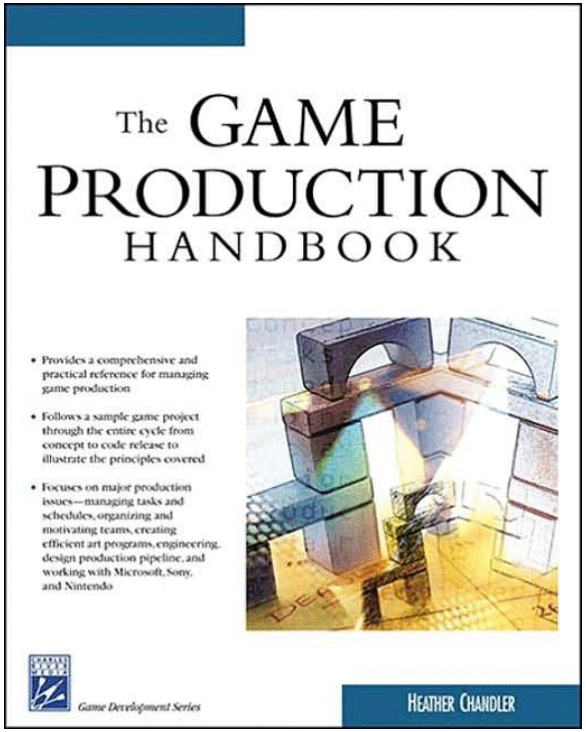
- Focus
  - Game implementation
  - Technical issues



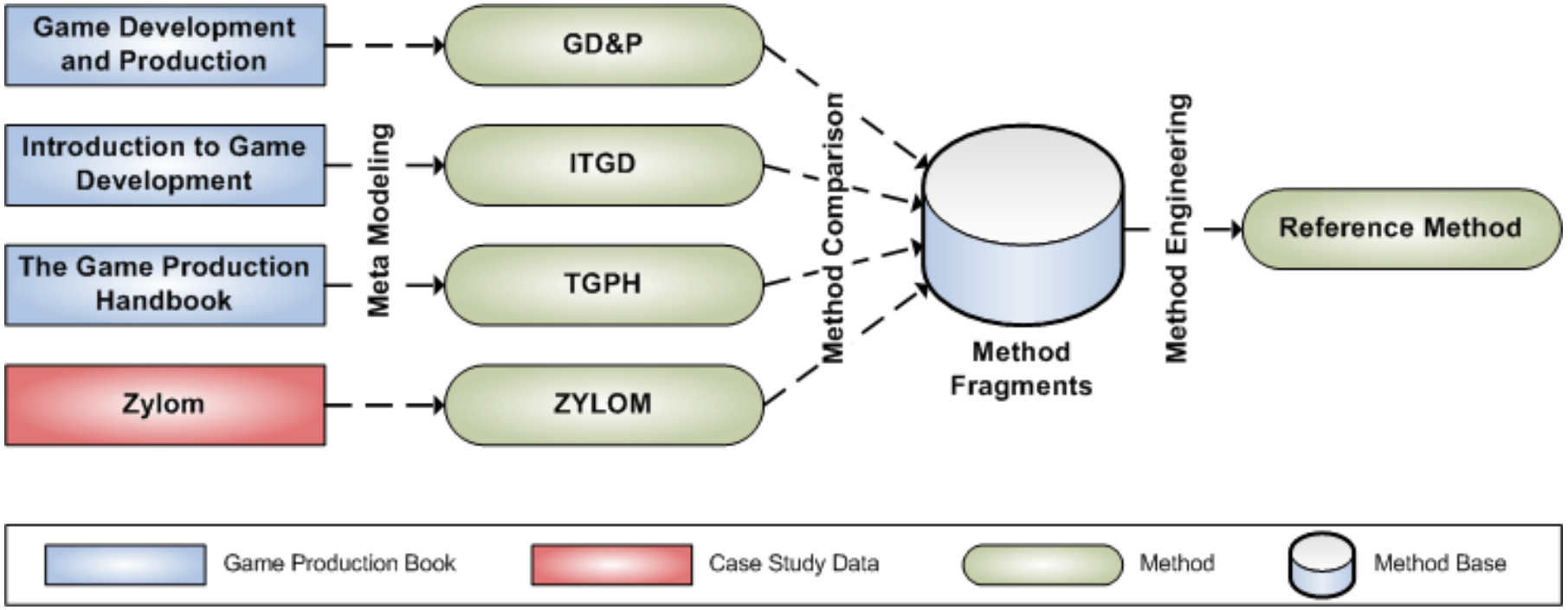
The game production handbook

Heather Chandler, 2006

- Focus
  - Development process
  - Templates and deliverables

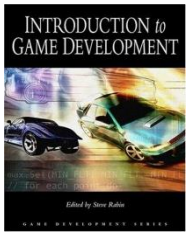


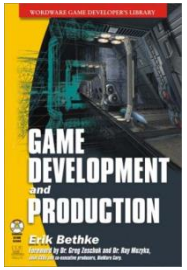
Approach

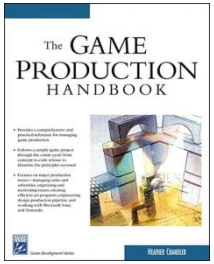

Game Development and Production
Introduction to Game Development
The Game Production Handbook
Zylom
Meta Modeling
GD&P
ITGD
TGPH
ZYLOM
Method Comparison
Method Fragments
Method Engineering
Reference Method
Game Production Book
Case Study Data
Method
Method Base

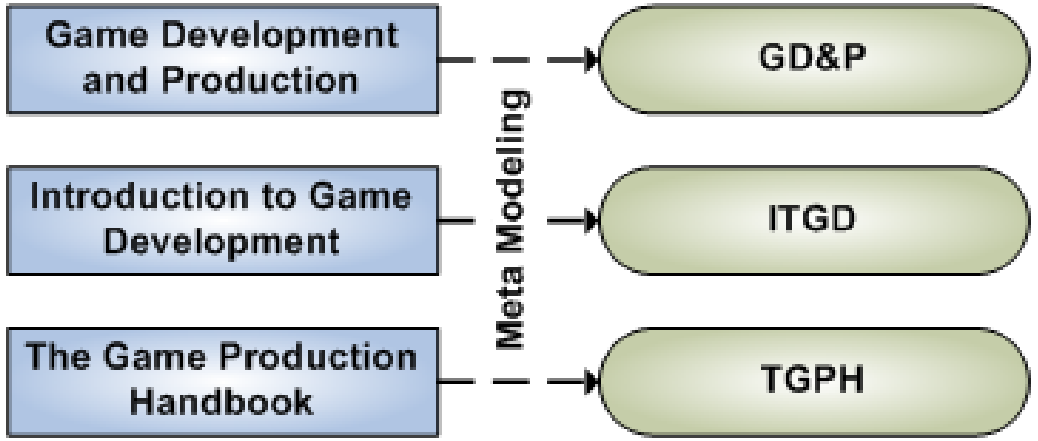
Literature analysis

- 3 books on Game Production
  - Translate theory into process deliverable diagrams using meta modeling

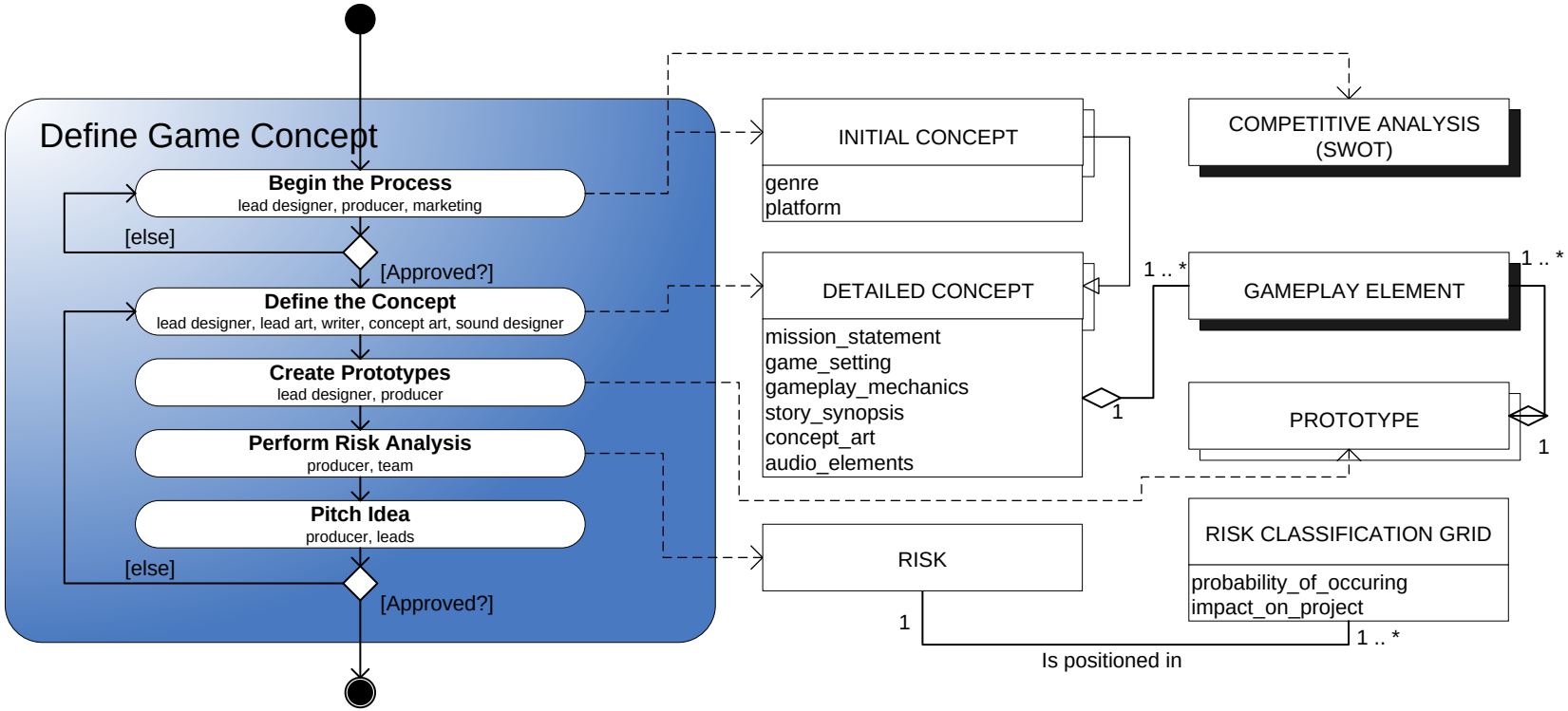








TGPH – Process-deliverable diagram



Process-deliverable diagram of the Concept Phase in "The Game Production Handbook"

TPGH – Activity table

Activity	Sub-Activity	Description
Define Game Concept	Begin the Process	At the start of the process, the Lead designer, Producer and Marketing manager develop an INITIAL CONCEPT for the new game. In the INITIAL CONCEPT the genre and platform on which the game is supposed to run are described. They also perform a COMPETITIVE ANALYSIS.
	Define the Concept	When the INITIAL CONCEPT is approved, the Lead designer, Lead art, Writer, Concept art, and Sound designer define the concept in a DETAILED CONCEPT.
	Create Prototype	The Lead designer and Producer create a PROTOTYPE, based on the DETAILED CONCEPT.
	Perform Risk Analysis	The Producer develops together with the rest of the Team a RISK CLASSIFICATION GRID, in which all RISKS are plotted.
	Pitch Idea	The Producer and Lead pitch the idea to the management. When it is approved, they can carry on with defining the game requirements.

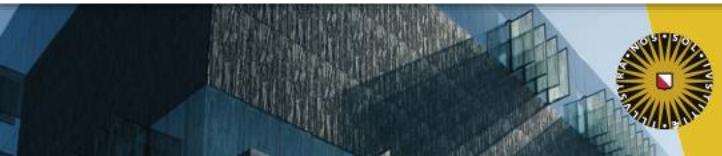
TPGH – Concept table

Concept	Definition
INITIAL CONCEPT	A not too detailed concept of the game that needs to present a compelling goal for the game to achieve.
DETAILED CONCEPT	A definition of the concept of a game that specifies the game mechanics, setting, characters, storyline, and major features.
COMPETITIVE ANALYSIS (SWOT)	An identification of the strengths and weaknesses of your game's competition, market opportunities for your game, and any threats that might impact the game's success in the market.
GAMEPLAY ELEMENT	A type of information defined in the detailed concept, like mission statement, game setting, gameplay mechanics, story synopsis, concept art, and audio elements.
PROTOTYPE	An original type, form, or instance serving as a basis or standard of the full game for later stages.
RISK CLASSIFICATION GRID	The result of performing risk analysis, where risks have been identified, analyzed and classified on probability of occurring and impact on the project.

Results

- Method fragments stored in method base
- Literature analysis statistics:

	GD&P	**ITGD**	**TGPH**
Activities	6	10	9
Sub activities	23	37	36
Concepts	36	42	49



Case study

Partner: Zylom Media Group B.V.

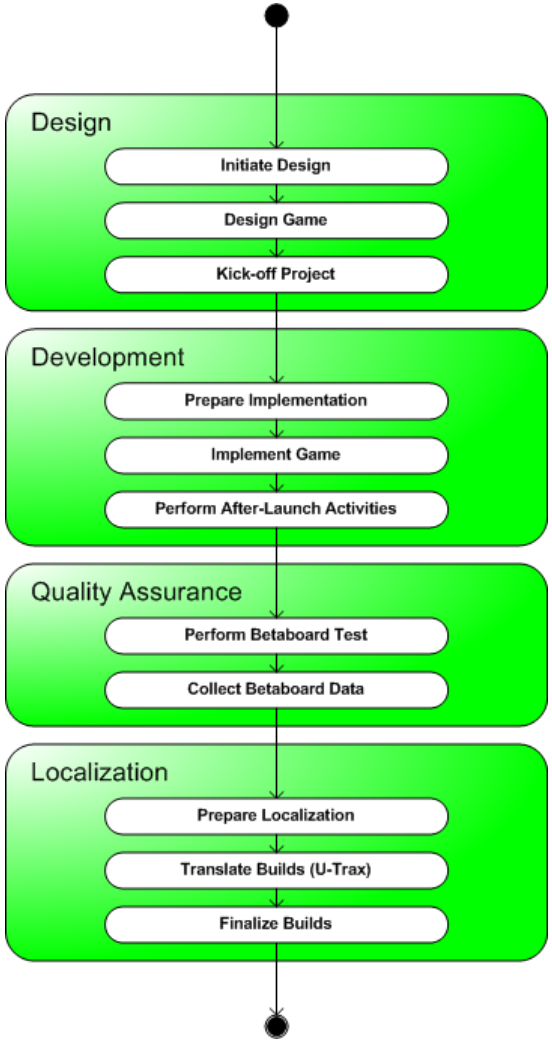
- Casual games
  - Development
  - Publishing
- Founded in 2001, Eindhoven
- More than 60 employees
- 10 M unique visitors a month


zylom
Have fun.
Andere taal / Other language
FUNPASS
SPELEN
DOWNLOADEN
HELP
Alle spellen
Kaart
Puzzel
Woord
Mahjong
3-op-een-rij
Zoek en Vind
Actie
Speel en Win
Lid worden | Inloggen | Inlognaam en/of wachtwoord vergeten?
Online spelletjes en download spellen, allemaal gratis bij Zylom. Probeer onze beste spellen meteen!
71362 spelers nu online
Gardenscapes
meer info
download
Binnenkort: Mortimer Beckett 3!
Gardenscapes Deluxe
Zoek en Vind
Onze populairste spellen
Ga de Zylom SuperMarkt binnen
Geniet met Zylom
Speel gratis online spelletjes
ACTIE
Gemini Lost
download
Farm Frenzy 3
download
ZOEK EN VIND
Gardenscapes
download
Big City Adventure -
download
FunPass
Een abonnement vanaf €9,95 per maand
Speel onbeperkt alle down-loadspellen

Case study approach

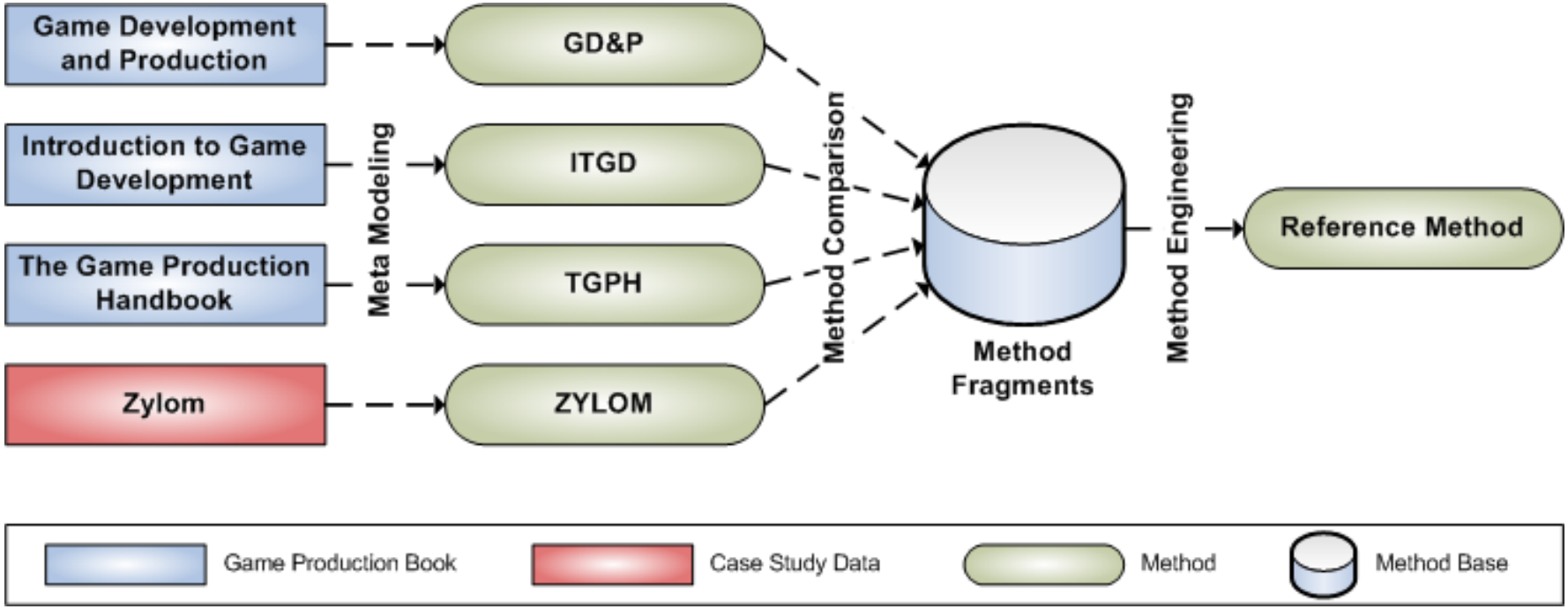
- 15 Interviewees
  - Collecting information
  - Reviewing draft versions
- Company documents
- Case study tactics (Yin, 2003)
- Information → Process deliverable diagrams

Case study details



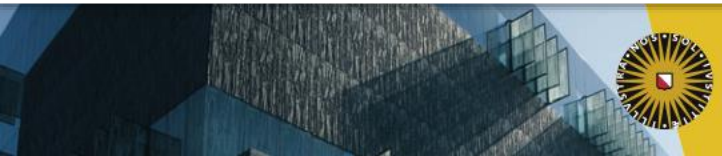
- 4 phases:
  design, development, quality assurance and localization
- 11 activities
- 59 sub activities
- 60 concepts

Approach


Game Development and Production
Introduction to Game Development
The Game Production Handbook
Zylom
Meta Modeling
GD&P
ITGD
TGPH
ZYLOM
Method Comparison
Method Fragments
Method Engineering
Reference Method
Game Production Book
Case Study Data
Method
Method Base

Comparison

- Define ‘super-method’
  - All activities and concepts in books and at Zylom
- Compare all four methods to super-method
  - Activity from super-method is available?
  - Activity from super-method does more or less?
  - Concept from super-method is available?
  - Concept from super-method has same name?



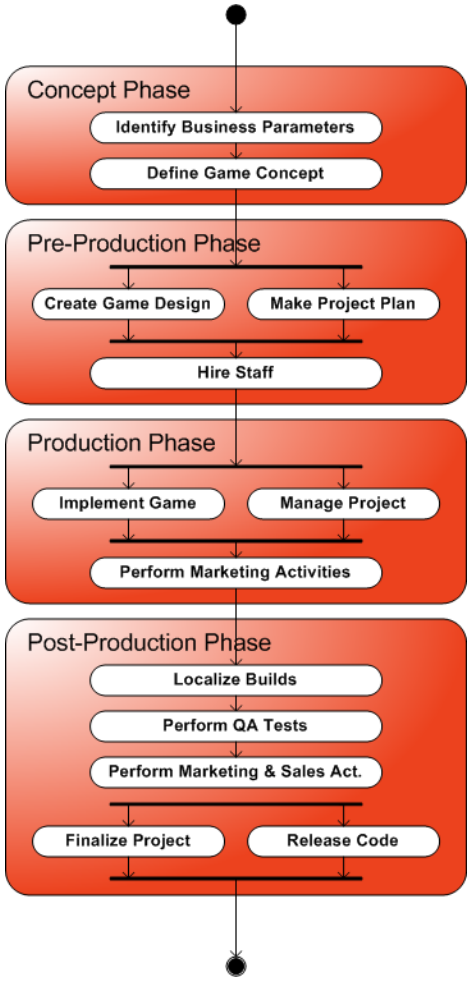
Activity comparison

2. Preproduction Phase	GD&P	ITGD	TGPH	Zylom
2.1 Create game design				
2.1.1 Brainstorm	=			
2.1.2 Delegate design	=			
2.1.3 Write game design document	=	>		=
2.1.4 Evaluate game design document				=
2.1.5 Write technical design document	=	><		<
2.1.6 Create visualizations				=
2.1.7 Present game design				=
2.1.8 Evaluate technology			=	
2.1.9 Define tools and pipeline			=	
2.1.10 Create documentation			=	

Concept comparison

2. Preproduction Phase	GD&P	ITGD	TGPH	Zylom
2.1 Create game design				
2.1.1 GAME DESIGN DOCUMENT	=	=	DESIGN DOC.	=
2.1.2 CORE GAMEPLAY	=			
2.1.3 CONTEXTUAL GAMEPLAY	=			
2.1.4 STORY	=			
2.1.5 TECHNICAL DESIGN DOCUMENT	=	=	TECHNICAL DOC.	=
2.1.6 REQUIREMENT	ASSET	FEATURE	FEATURE	ASSET
2.1.7 VISIBLE REQUIREMENT	=			
2.1.8 NONVISIBLE REQUIREMENT	=			
2.1.9 FEATURE LIST			=	ASSET LIST
2.1.10 DOCUMENTATION			=	
2.1.112 PROTOTYPE				=

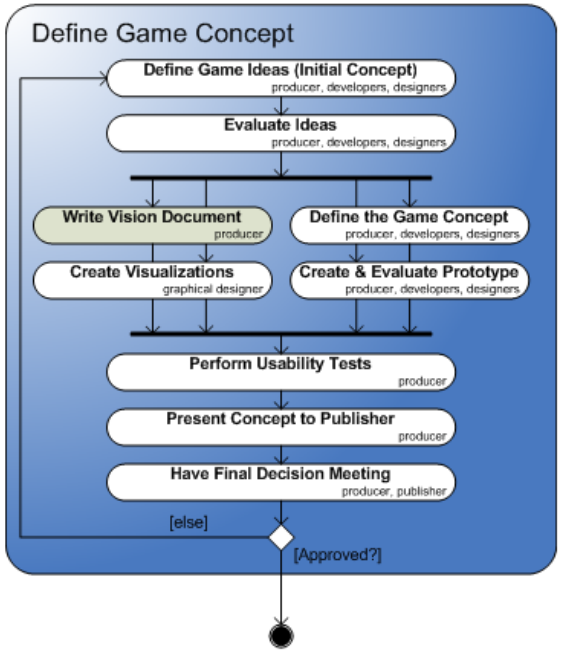
Reference method



- Complete process
- Selection of all *relevant* method fragments
- So, the reference is a subset of the supermethod, which can be tailored to fit each situation.
- Result:
  - 13 main activities
  - 69 sub activities
  - 93 deliverables

Game Production Reference Method

Universiteit Utrecht



Description

Write Vision Document results in a VISION DOCUMENT

For more information, see Game Development and Production (2003) Bethke, E. (Eds.)

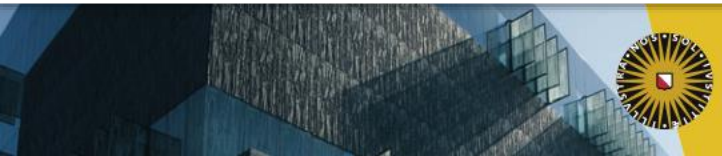
Deliverables

An executive summary of the game design document that touches upon all the key features of the game in such a manner to grab them and get them to request to see the game demo and move forward with a deal.

Conclusions

- Reference method for game production
  - Uniform terminology
- Process improvements for Zylom
- Improvements of method comparison approach (Hong et al., 1993) by using activity and concept tables
- Public knowledge infrastructure

QUESTIONS?



References

Jeusfeld, M.A., M. Jarke, & Mylopoulos, J. (2009*). Metamodeling for Method Engineering.* Camebridge, USA: The MIT Press.

Karlsson, F., & Ågerfalk, P. J. (2004). Method configuration: adapting to situational characteristics while creating reusable assets. *Information and Software Technology*, 46(9), 619-633.

Hong, S., Goor, G. van den, Brinkkemper, S. (1993). A formal approach to the comparison of object-oriented analysis and design methodologies. Proceedings of the 26th Annual Hawaii International Conference on System Sciences, Hawaii, 689–699.

Ralyté, J., Deneckère, R., & Rolland, C. (2003). Towards a generic model for situational method engineering. *Lecture Notes in Computer Science 2681*, 95-110.

Kevin Vlaanderen, Sjaak Brinkkemper, Inge van de Weerd: On the Design of a Knowledge Management System for Incremental Process Improvement for Software Product Management. *Int J Information Systems Modeling and Design* 3(4): 46-66 (2012)