MSF-based process patterns

Dmitry Malenko (dmalenko@acm.org)
Vladimir Pavlov (vlpavlov@ieee.org)

About the authors

- **Dmitry Malenko** (Ukraine)
  - Graduate student, Dnepropetrovsk National University
  - MCSD for .NET
  - Member of ACM
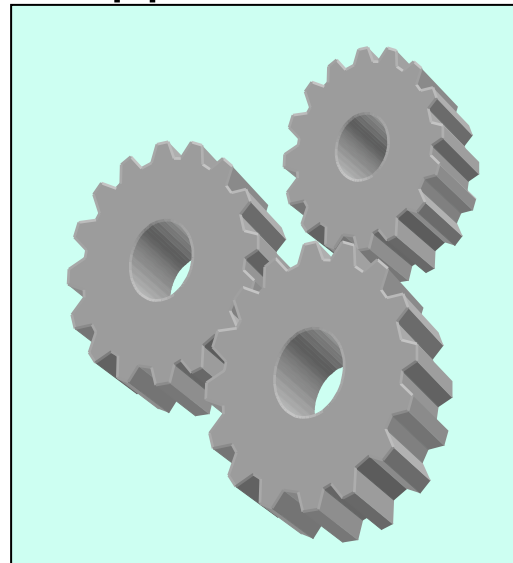
- **Vladimir Pavlov** (Ukraine/USA)
  - Chief Technical Officer of eLine Software, Inc.
  - Microsoft Endorsed MSF Practitioner, MCSD for .NET, MCSD, MCDBA, MCT, CompTIA Certified IT Project+
  - Member of PMI, ACM, IEEE and IEEE Computer Society
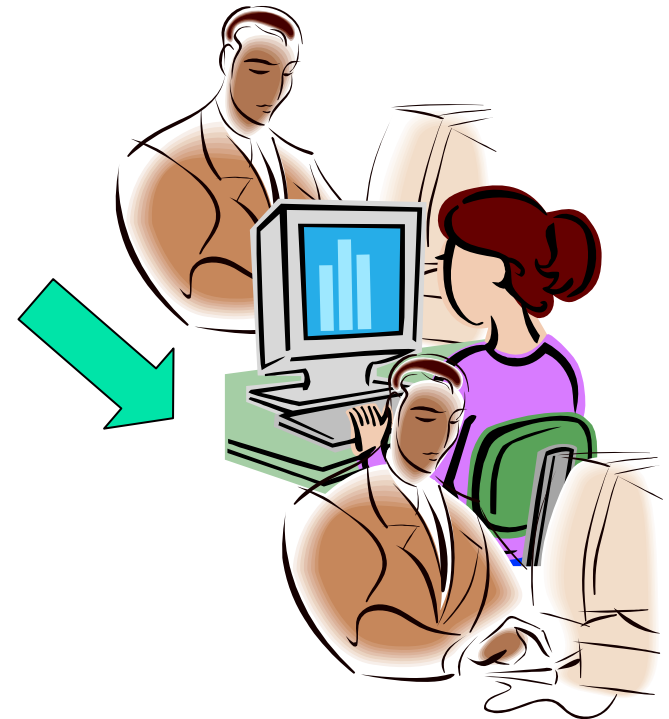
SPEM and Process Engineering: the big picture
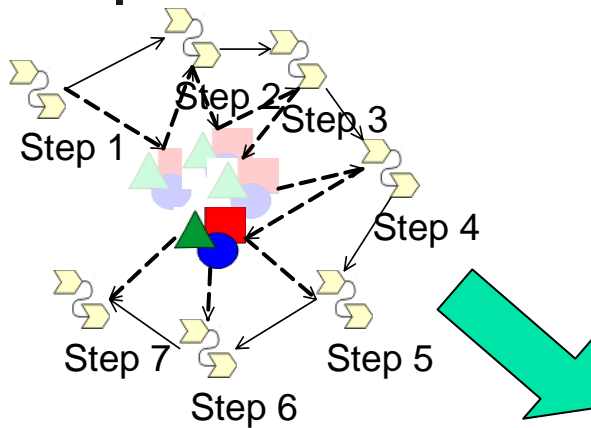
Tool that supports SPEM

Set of automated workbenches

SPEM Model

Our agenda

- **Introduction**
  - From design patterns to process and organizational patterns
  - From UML to SPEM
  - Microsoft Solutions Framework
- **Process patterns in MSF**
  - Living document
  - Reenterable process
  - Smart lifecycle
  - Stakeholder-oriented organization

Design patterns

Design patterns are descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context

E. Gamma at al. *Design Patterns: Elements of Reusable Object-Oriented Software*, 1995.

The way to design patterns

- Design *déjà-vu* — feeling that you've solved a problem before but not knowing exactly where or how
- Reusing and refining design solutions we had in past

Appeal of design patterns

- Depicted using UML notation
    - Easy to understand at one sight
    - Formal enough
- Clear rationale behind
- Highly abstract and therefore widely reusable
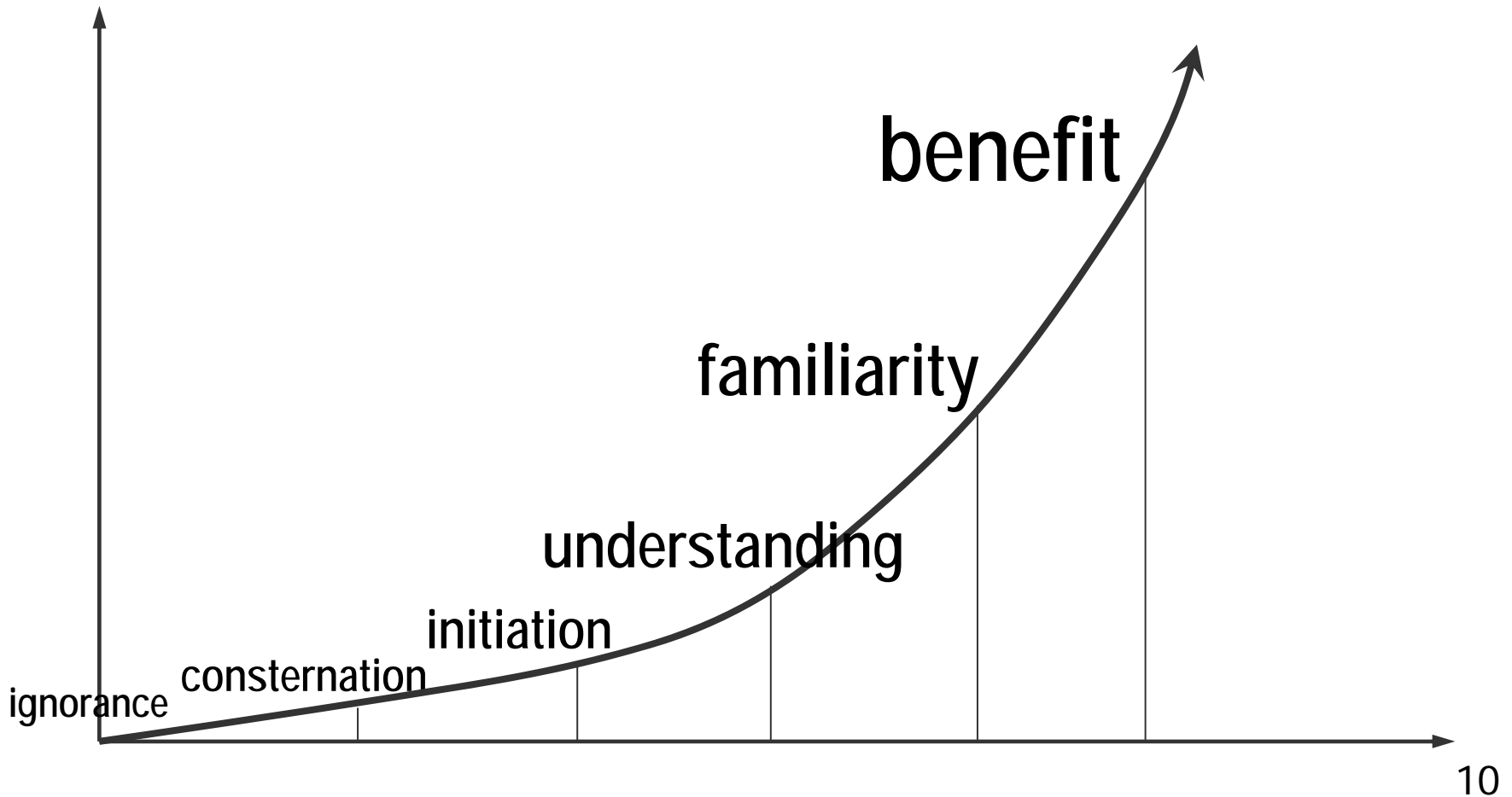
Popularity of design patterns

- Internet search for "design patterns" gives more than 7 000 000 links

- More than 300 books on the subject on the Amazon.com

- IDE support for design patterns at programmer's workspace (IBM Rational XDE)
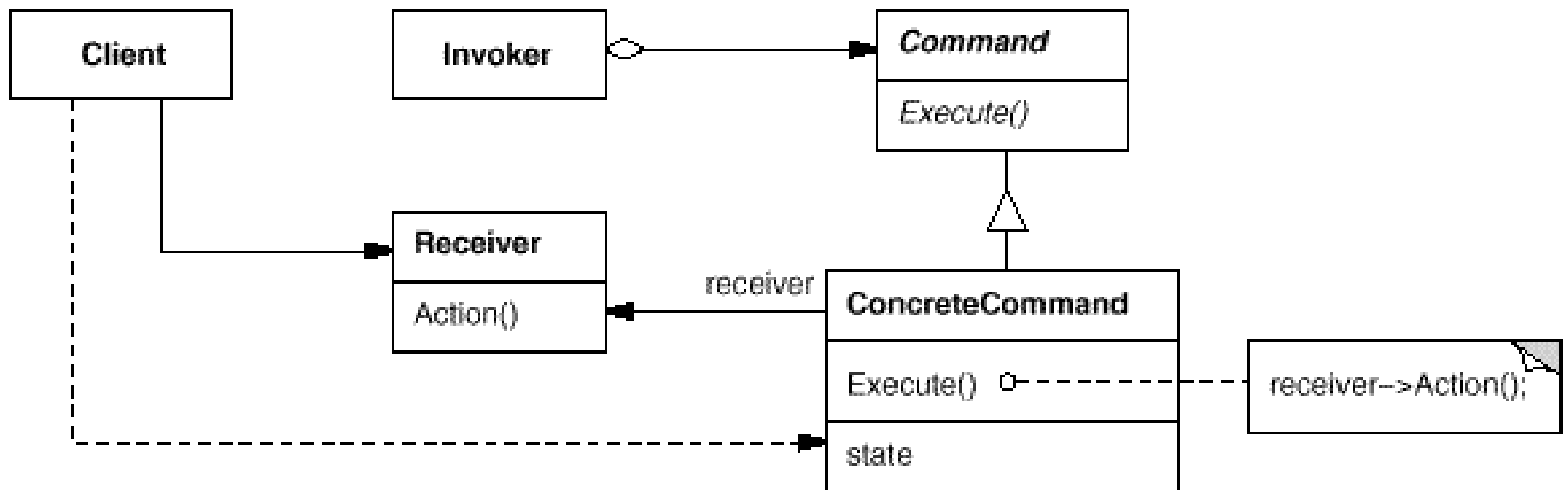
Design patterns community

- Workshops at major conferences
  - OOPSLA – http://oopsla.acm.org
  - ECOOP – http://www.ecoop.org
  - ICSE – http://www.icse-conferences.org
- Lots of publications in different journals
  - E.g. http://www.research.ibm.com/designpatterns/publications.htm
- Large Internet-community of users and contributors
  - http://hillside.net/patterns

Effectiveness does not come easy

benefit

familiarity

understanding

initiation

consternation

ignorance

Sample pattern: Command

Sample pattern: Command

- **Consequences**
  - Command decouples the object that invokes the operation from the one that knows how to perform it
  - Commands are first-class objects. They can be manipulated and extended like any other object
  - You can assemble commands into a composite command
  - It's easy to add new Commands, because you don't have to change existing classes

Way to process patterns

- Many different software processes were created. All of them tried to combine best practices with some fresh ideas

- Design patterns have proved their value and it was tempting to apply similar technique to software development processes

Process patterns

Process patterns describe a proven, successful approach and/or series of actions for developing software

First introduced by Jim Coplien's paper "A Generative Development-Process Pattern Language", 1994

Literature on process patterns

- Coplien, J.O. *A Generative Development-Process Pattern Language*. Pattern Languages of Program Design, Addison Wesley Longman, Inc., pp. 183-237, 1995
- Ambler, S. W. *Process Patterns: Building Large-Scale Systems Using Object Technology*. New York: SIGS Books/Cambridge University Press, 1998
- Ambler, S. W. *More Process Patterns: Delivering Large-Scale Systems Using Object Technology*. New York: SIGS Books/Cambridge University Press, 1998
- Pattern Languages of Program Design Series
- More on http://hillside.net/patterns/papersbibliographys.htm

Process patterns community

- **Conferences and workshops**
  - Conference on Pattern Languages of Programs - http://st-www.cs.uiuc.edu/~plop/
  - Workshop on Software Development Process Patterns at OOPSLA
    - http://oopsla.acm.org/fp/files/wor-16.html
    - http://wwwbib.informatik.tu-muenchen.de/infberichte/2002/TUM-I0213.pdf.gz
  - …
- **Large Internet-community of users and contributors**
  - http://www.ambysoft.com/processPatternsPage.html

Why process patterns?

- Process engineering benefits
- Documented experience of software development process organization
- Methodology agnostic – can be applied to whatever methodology you use
- Support for process patterns within process engineering tools in future

Sample pattern: Team per task (Alistair Cockburn, Risk management patterns catalog)

- **Sensation**
  - We are getting distracted here. We are losing precious design cycles
- **Symptoms**
  - The project is not moving forward, even though you think it should be. People have multiple things to do (as usual). A secondary task is dominating their time, keeping them from moving forward with their primary goal. Possibly, they are spending so much energy switching contexts they cannot get a clear mind to do their main task
- **Forces**
  - On the one hand, if you let them drop any of their tasks, your project will miss an important date. So you want several tasks moved forward at one time. On the other hand, having people active on these multiple tasks is not working well
- **Try This**
  - Split the team. Sort the activities so that each team has a primary task and set of activities that are not mutually distracting (designing software and sitting in meetings or answering phone calls are mutually distracting). Arrange it so that each team can focus on its primary task, and each task has a dedicated team member
- **Counterforce**
  - You will eventually have one-person teams. Before then, you may discover that it is not worth splitting up the task set the team has because the working synergy between people that is lost is more harmful than the dedicated time gained per task

Some observations

- Idea is difficult to capture from first sight
- Being too specific makes it harder to reuse
- Most of the times can be called "principle" rather than "pattern"
- No formal description

From UML to SPEM

- UML is an industry standard for OO modeling
  - http://www.omg.org/technology/documents/formal/uml.htm
- "Pure" UML is not perfect for software process modeling
- SPEM – Software Process Engineering Meta-model
  - http://www.omg.org/technology/documents/formal/spem.htm

Software Process Engineering Meta-model

- SPEM is an ordinal UML profile that aids software process modeling
- Process structure
  - WorkProduct, WorkDefenition, Activity, Step
- Process components
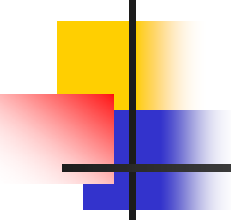 - Process, Discipline
- Process lifecycle
  - Phase, Iteration

Conceptual Model
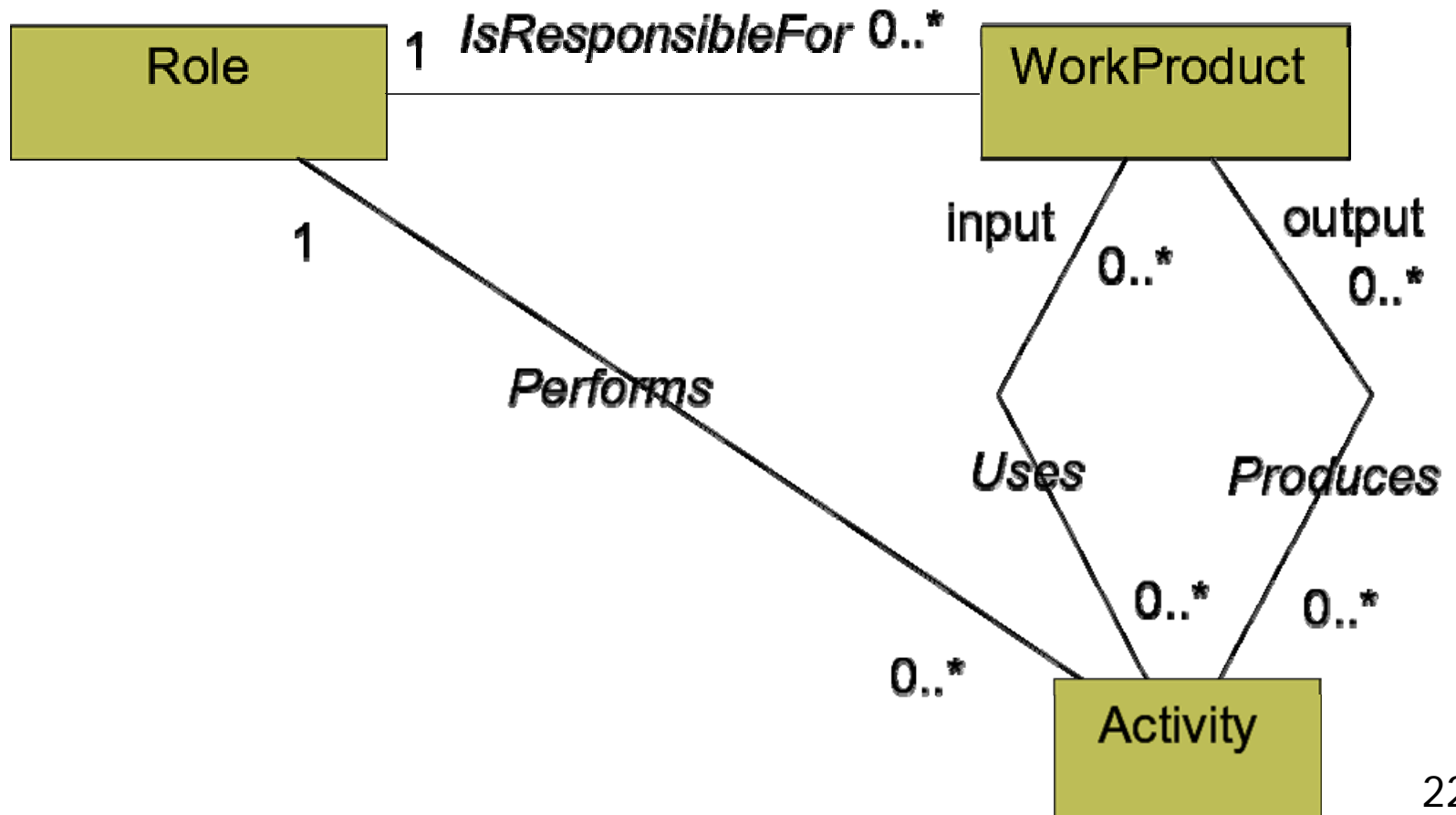
Why SPEM?

- Standardizes a way of expressing *any* software development process

- Can be used with *any* methodology, tool, framework

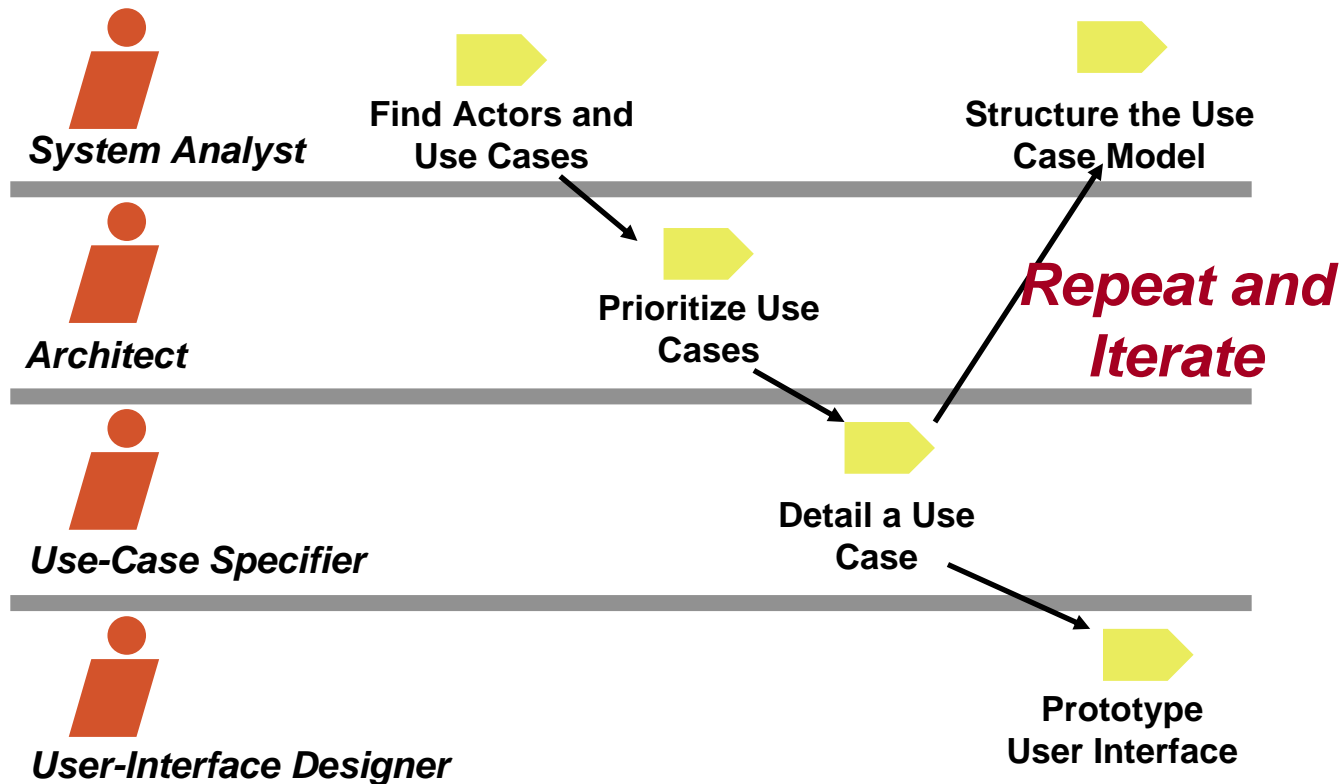- Leverages expressiveness and popularity of UML

Processes that have SPEM-base definition

- Rational Unified Process
- DMR Macroscope
- IBM's Global Services Method
- Unisys QuadCycle
- …

Sample SPEM diagram from Rational Unified Process
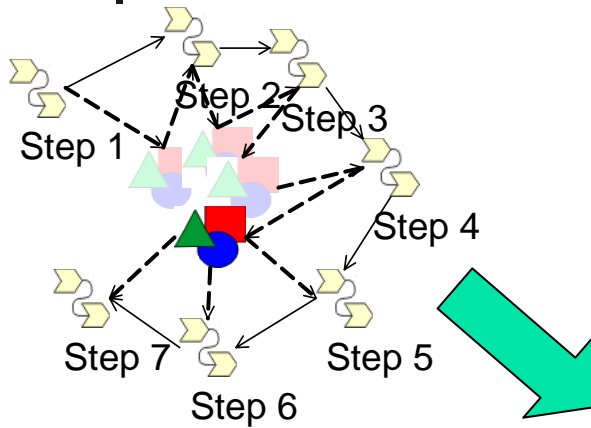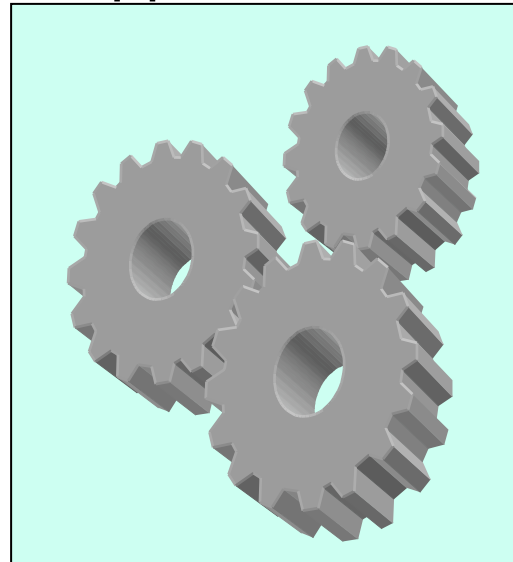
Requirements Core Workflow

System Analyst

Find Actors and Use Cases

Structure the Use Case Model

Architect

Prioritize Use Cases

Repeat and Iterate

Use-Case Specifier

Detail a Use Case

User-Interface Designer

Prototype User Interface

Tools that support SPEM

- Iris
  - Can be used to enact designed process
  - http://www.osellus.com/products/iris.html
- IBM Rational Process Workbench
  - Can be used for RUP customization
  - http://www.ibm.com/software/awdtools/rup/workbench
- Objecteering/UML
  - Supports SPEM Profile
  - http://www.objecteering.com/products.php
- Enterprise Architect
  - Supports SPEM Profile
  - http://www.sparxsystems.com.au/ea.htm

- More to appear…

Process engineering
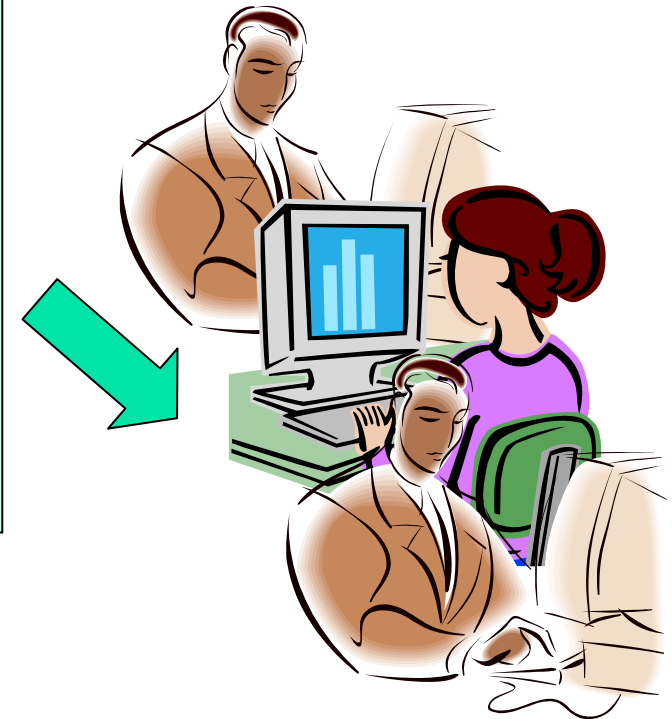
Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
Step 7

SPEM Model

Tool that
supports SPEM

Set of automated
workbenches

Process patterns applied

- Process engineer creates a process by combining best practices
- Software development projects face some common problems – common solutions exist
- Process patterns leverage previous experience
- Formal definition of pattern allows for easier incorporation of known solution into new process

Future of SPEM

- SPEM 2.0 is being worked on
- Alignment with Business Process Definition Meta-model
- Process modeling tools that support SPEM begin to emerge
- More and more organization use SPEM to model and engineer their processes

Process patterns and SPEM

- SPEM can be used for description of process patterns like UML is used for description of design patterns
- SPEM allows for more formal definition of process and organizational patterns
- Process patterns can be described in Gamma-style

Microsoft Solutions Framework

- The Microsoft Solutions Framework (MSF) is a collection of Microsoft's proven practices on managing successful IT projects

- Microsoft almost does not market MSF and they are not selling it, instead, they focus on making money using MSF

- Similar to Windows or any other product, MSF evolves and matures as long as new versions are released. Initially Microsoft made MSF available in 1994. The latest version of MSF is 3.0

- You cannot buy this product from Microsoft, but you still can get it – it's free. Both whitepapers describing MSF and sample templates for MSF lifecycle deliverables are available on the Microsoft web-site
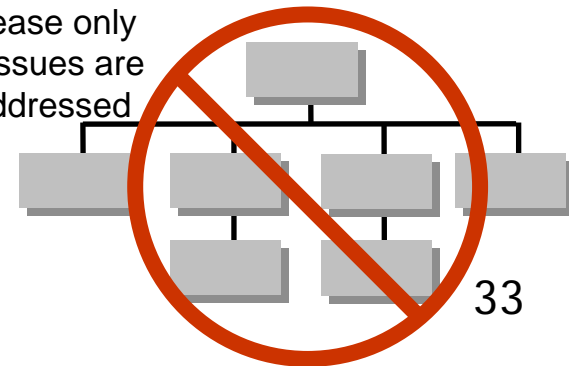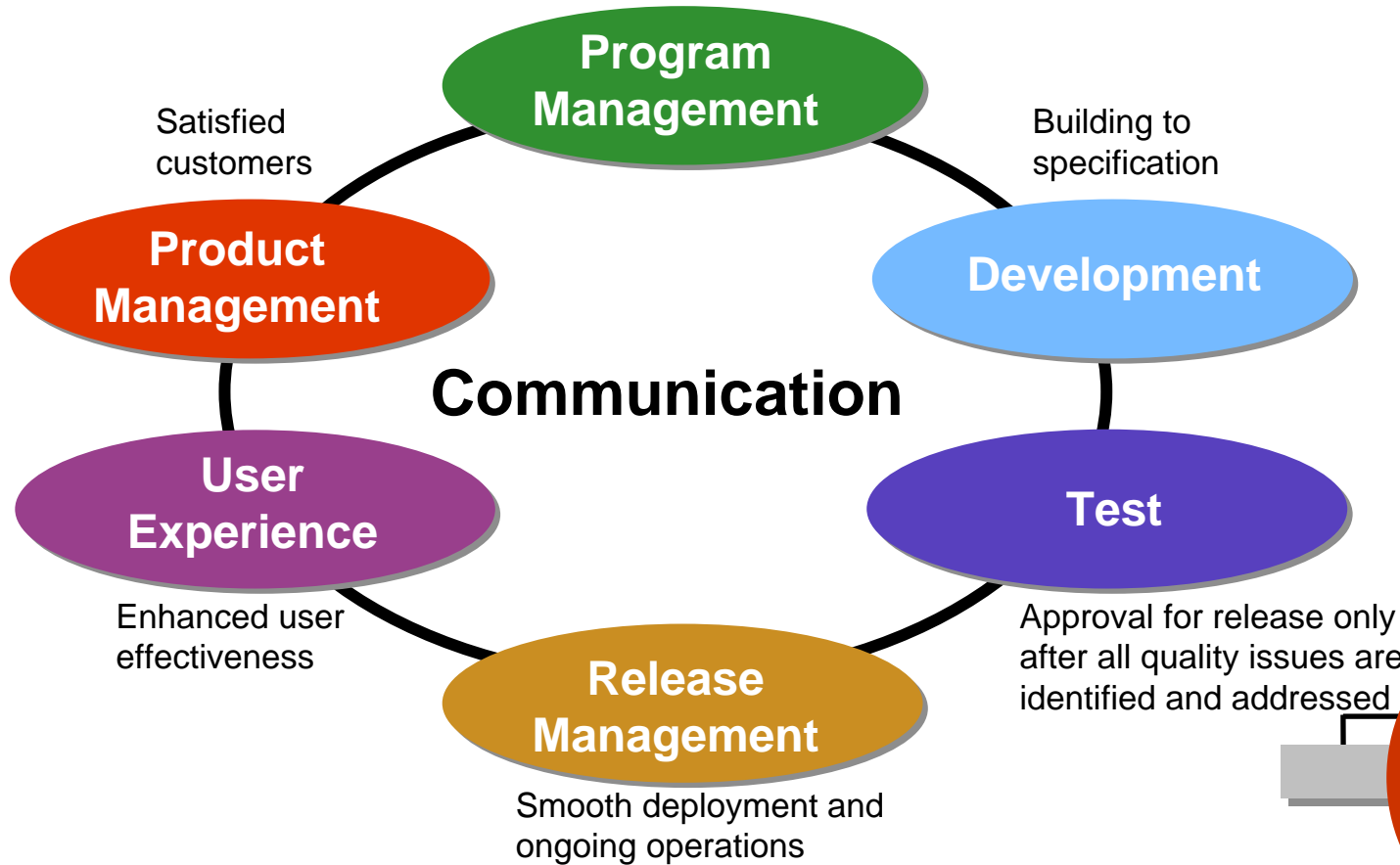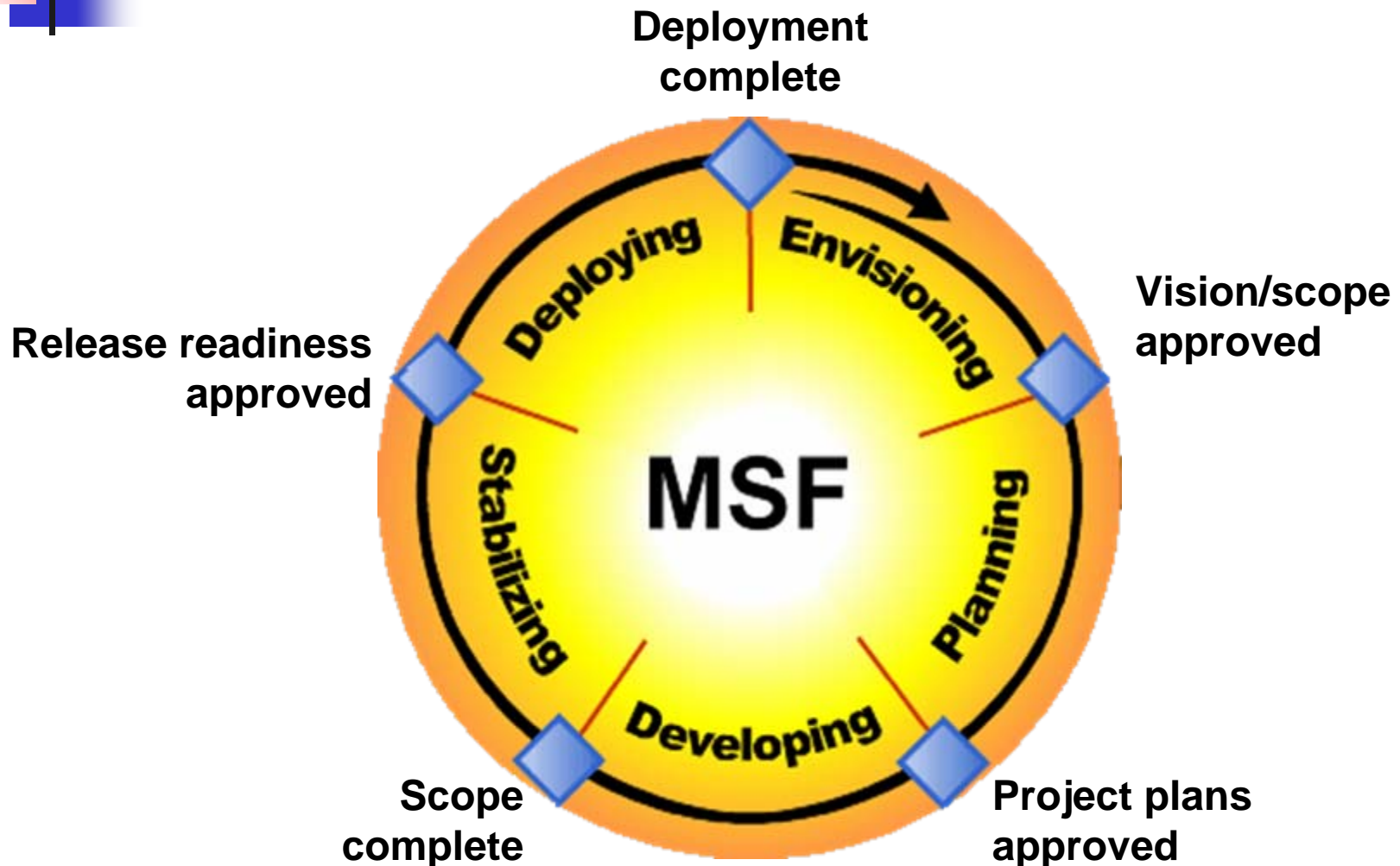
Microsoft Solutions Framework

- Software process used within Microsoft
  - http://www.microsoft.com/msf (English)
  - http://www.microsoft.com/rus/msf (Russian)
- Microsoft Solutions Framework consists of:
  - MSF Team Model
  - MSF Process Model
  - MSF Project Management Discipline
  - MSF Risk Management Discipline
  - MSF Readiness Management Discipline

MSF Team Model

Delivering the solution
within project constraints

Program Management

Satisfied customers

Building to specification

Product Management

Development

Communication

User Experience

Test

Enhanced user effectiveness

Approval for release only
after all quality issues are
identified and addressed

Release Management

Smooth deployment and
ongoing operations

33

MSF Process Model

Deployment complete

Vision/scope approved

Release readiness approved

Scope complete

Project plans approved

Deploying

Envisioning

Stabilizing

MSF

Planning

Developing

MSF Project Management Discipline

A bridge between MSF and PMBOK

Team Leads	Integration Management	Scope Management	Time Management	Cost Management	Communications Management	Communications Management	Risk Management	Procurement Management	Quality Management
Program Management	●	●	●	●	●	●	●	●	●
Product Management	○	○	○		○	●	○		○
Development	○	○	○		○	○	○		○
Test	○	○	○		○	○	○		○
User Experience	○	○	○		○	○	○		○
Release Management	○	○	○		○	○	○	●	○

● at overall project level   ○ at sub-team level
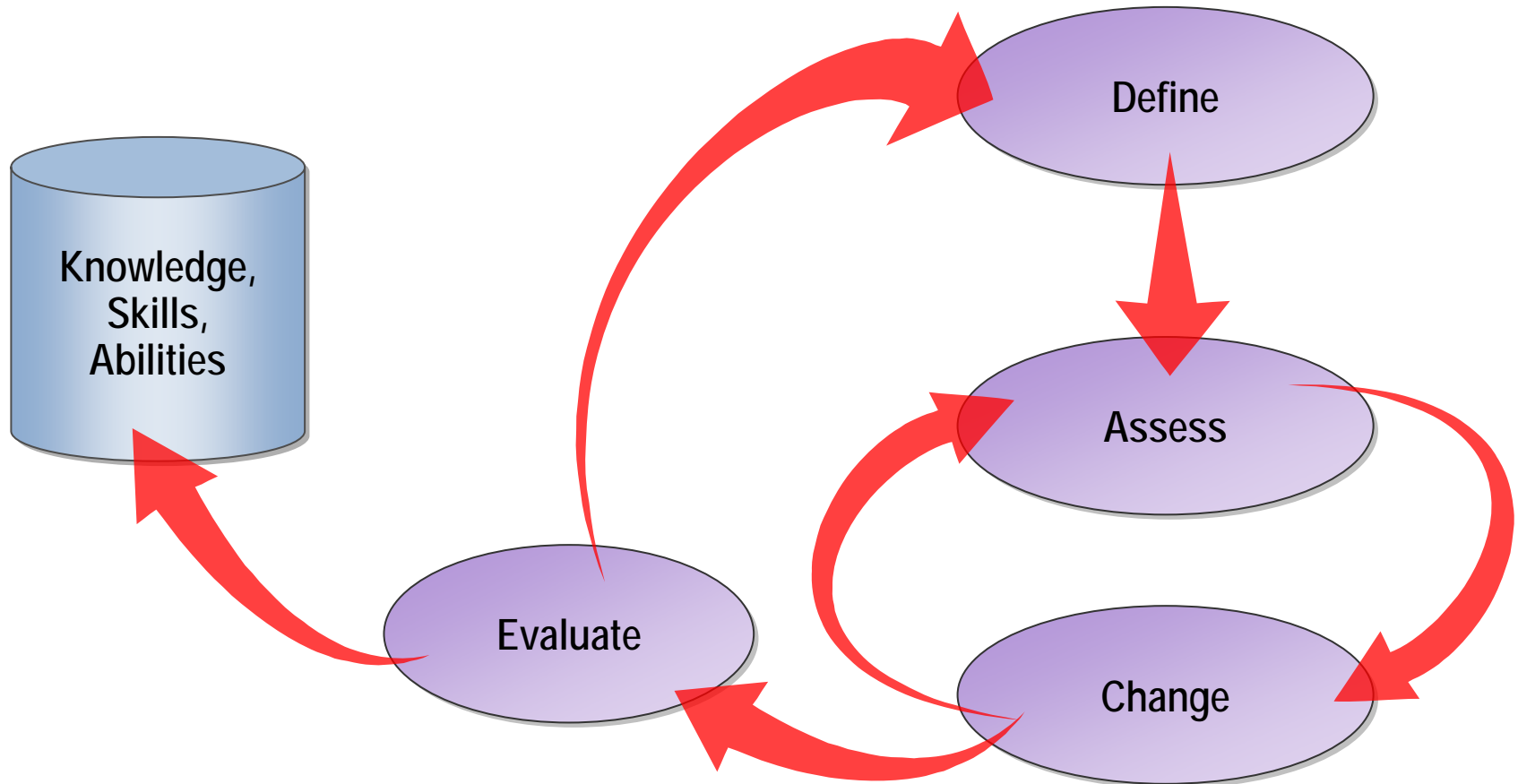
35

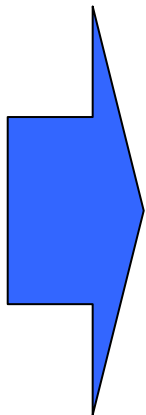# MSF Risk Management Discipline

MSF Readiness Management Discipline

Our agenda

- Introduction
    - From design patterns to process and organizational patterns
    - From UML to SPEM
    - Microsoft Solutions Framework
- Process patterns in MSF
    - Living document
    - Reenterable process
    - Smart lifecycle
    - Stakeholder-oriented organization

MSF in SPEM

- Many software processes were described using SPEM

- Formal definition of the process can be used to generate tool for supporting that process

- MSF is an "agile" process – can SPEM be used to model "agile" processes?

Our project

- Create a formal description of MSF in SPEM
- Find out whether SPEM is applicable to modeling of "agile" processes
- Extract process patterns from MSF model in SPEM
- Give general description of discovered process patterns

Our project

- **Started in December 2003**
- **We used Enterprise Architect with SPEM Profile to create a model of MSF**
  - About 20 diagrams define project team structure and different disciplines/processes
  - More than 10 ProcessRoles and more than 15 ProcessPerformers associated with more than 20 WorkDefinitions

- **A separate report on SPEM model of MSF is being worked on**

The same process from different points

- Different diagrams can be used to describe same process
- Not only shown transitions are allowed
- Different perspectives of same process give better understanding

Patterns discovered

- Living document
- Reenterable process
- Smart lifecycle
- Stakeholder-oriented organization

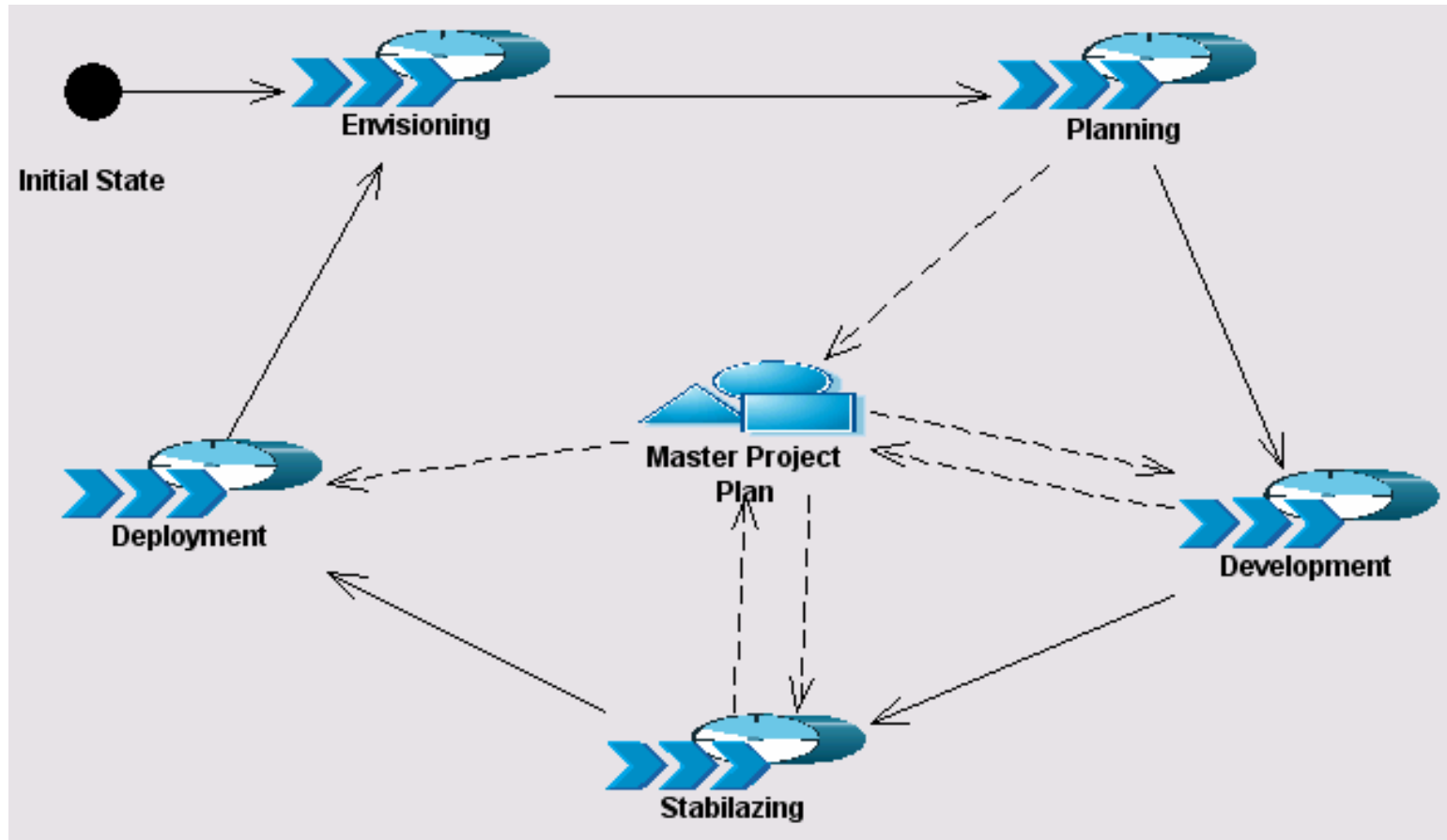- Other MSF-based patterns are to be discussed in further presentations …

MSF-based process patterns

- Living document

- Reenterable process
- Smart lifecycle
- Stakeholder-oriented organization

MSF Project Lifecycle

Master Project Plan

This should be 3D-diagram. Why? Take a look at slides 46-47

MSF Risk Management Discipline

This should be 3D-diagram. Why? Take a look at slides 46-47

Initial State

Risk Analysis and Prioritization

Risk Planning and Scheduling

Risk Identification

Master Risk List

Risk Tracking and Reporting

Risk Learning

Risk Control
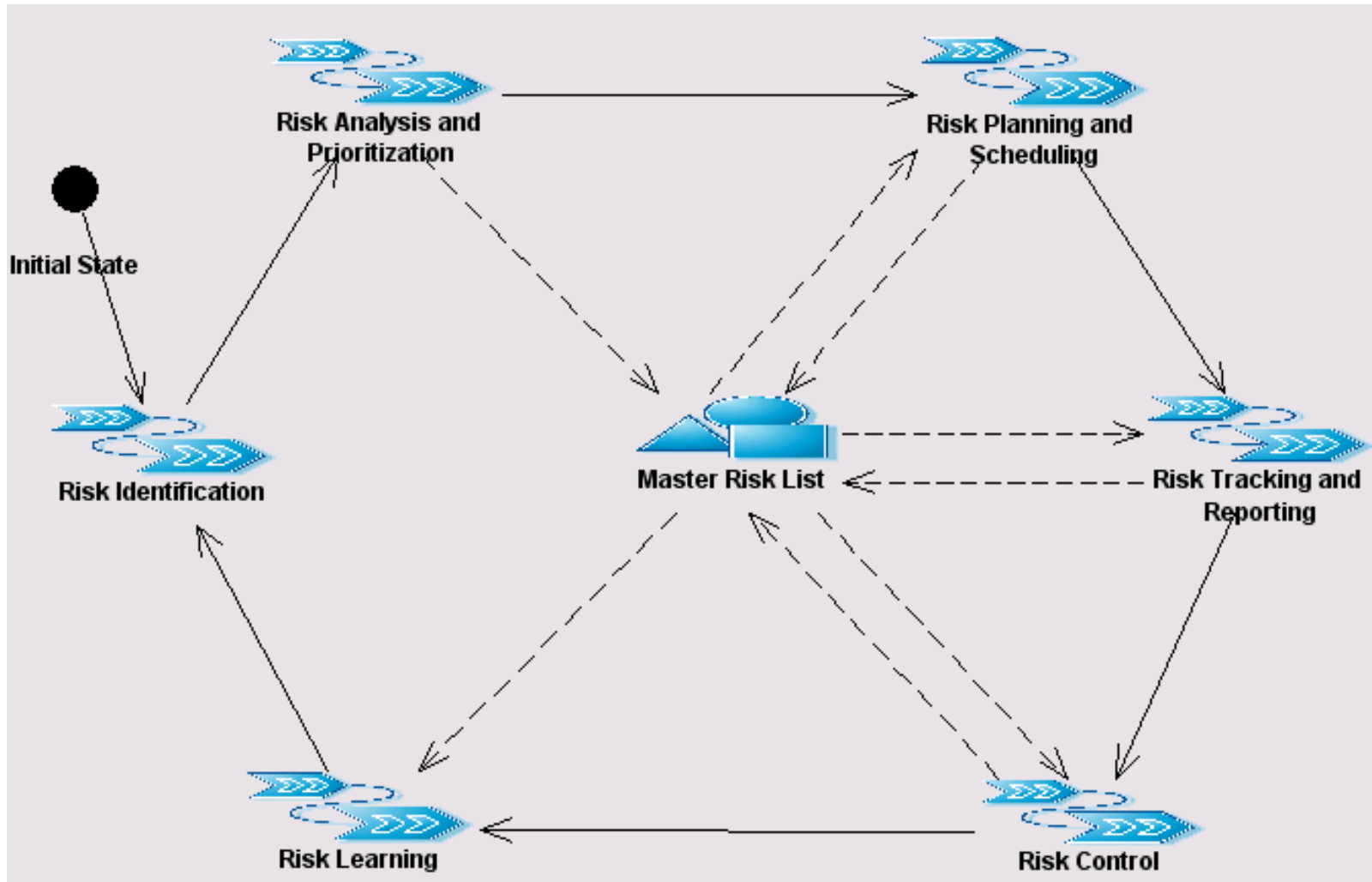
Déjà-vu – we've already seen that...

- Last two diagrams look very similar – it is reasonable to expect common rationale behind them

- It is a pattern!!!

- A closer look at the models we created allowed us to identify some more patterns
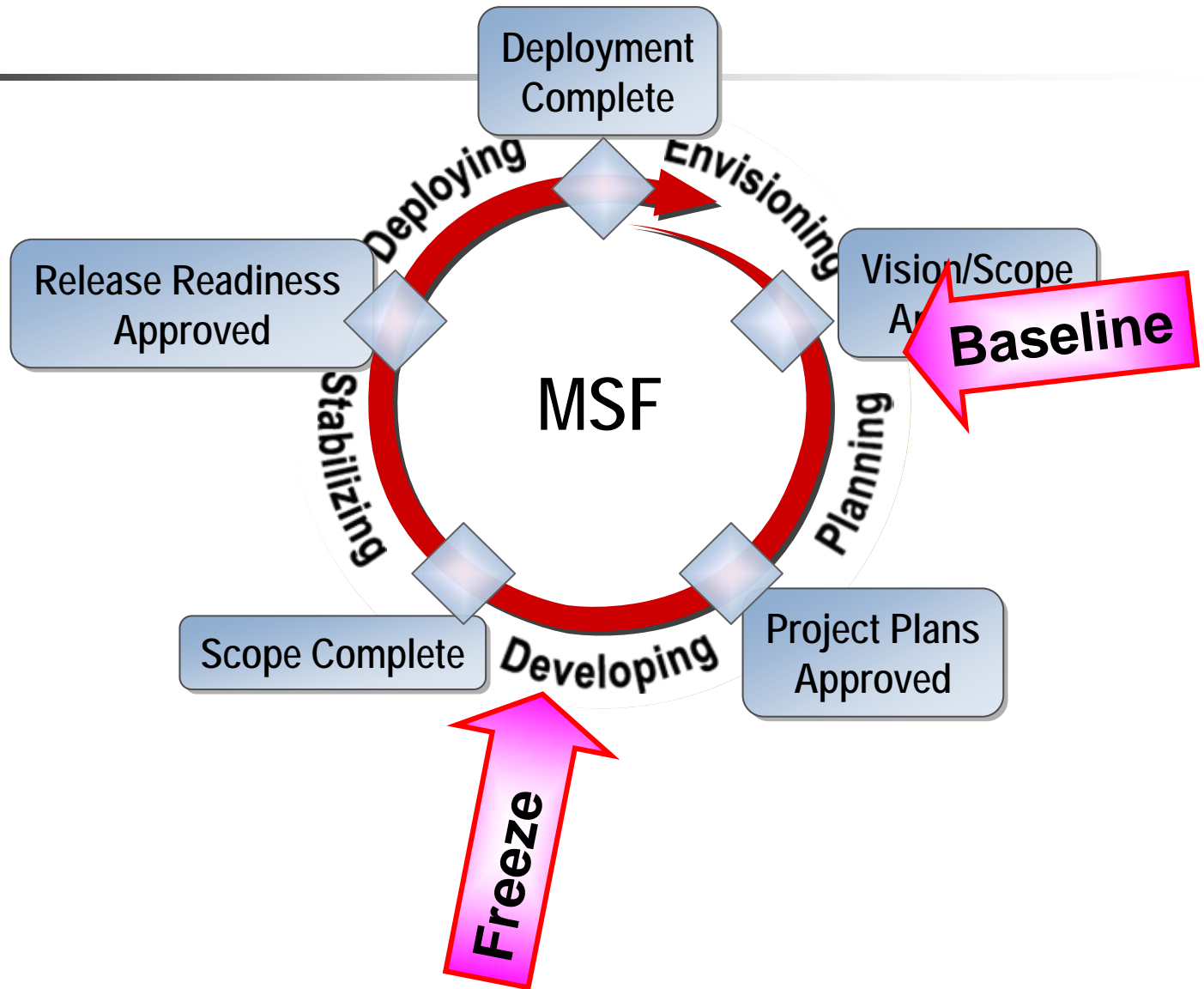
Living document (UML)

Living document (SPEM)

Step 2

Step 3

Step 1

Document

Document

Document

Document

Step 4

Step 7

Step 5

Step 6
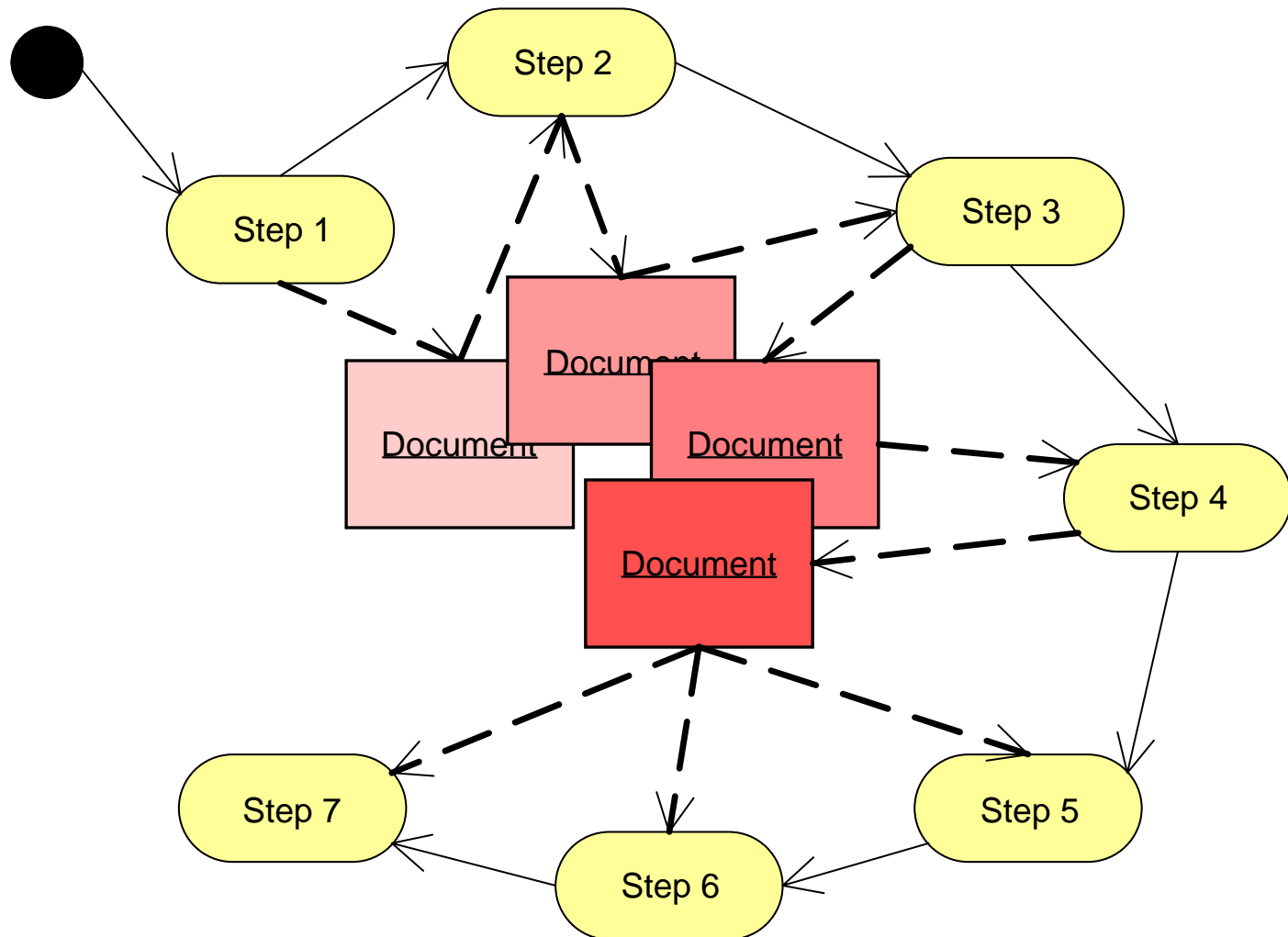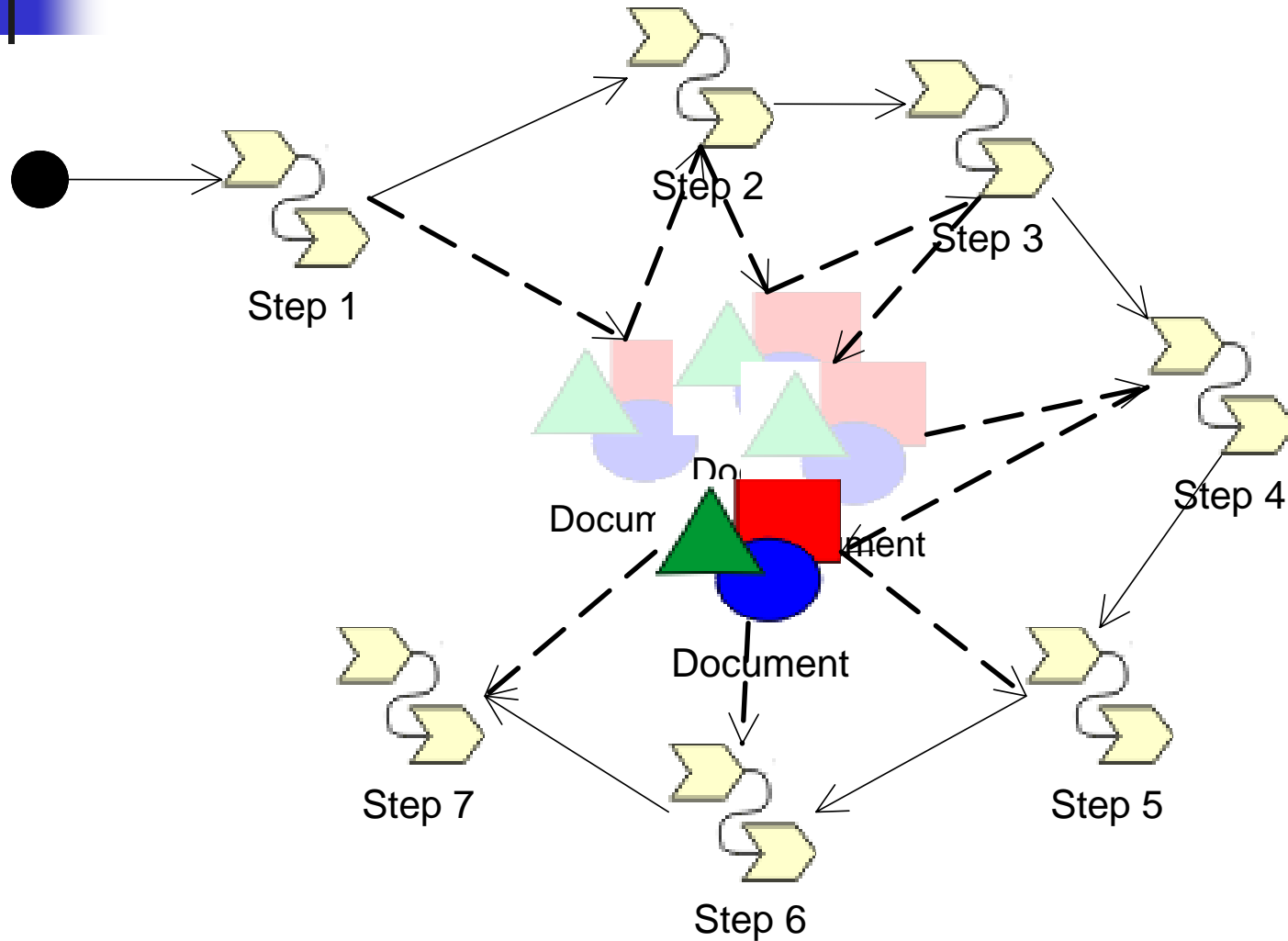
Living document

- Intent
  - To make sure important decisions can be baselined as early as possible and frozen as late as possible
- Motivation
  - Very often document created once as output of certain step of process is seen as stale and changes to the document are not appreciated. This prevents project team from effectively incorporating latest information that affect decisions made earlier

Rationale

- MSF
  - Living Document
  - Baseline early, freeze late
- Agile
  - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage
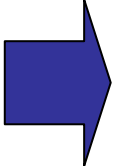
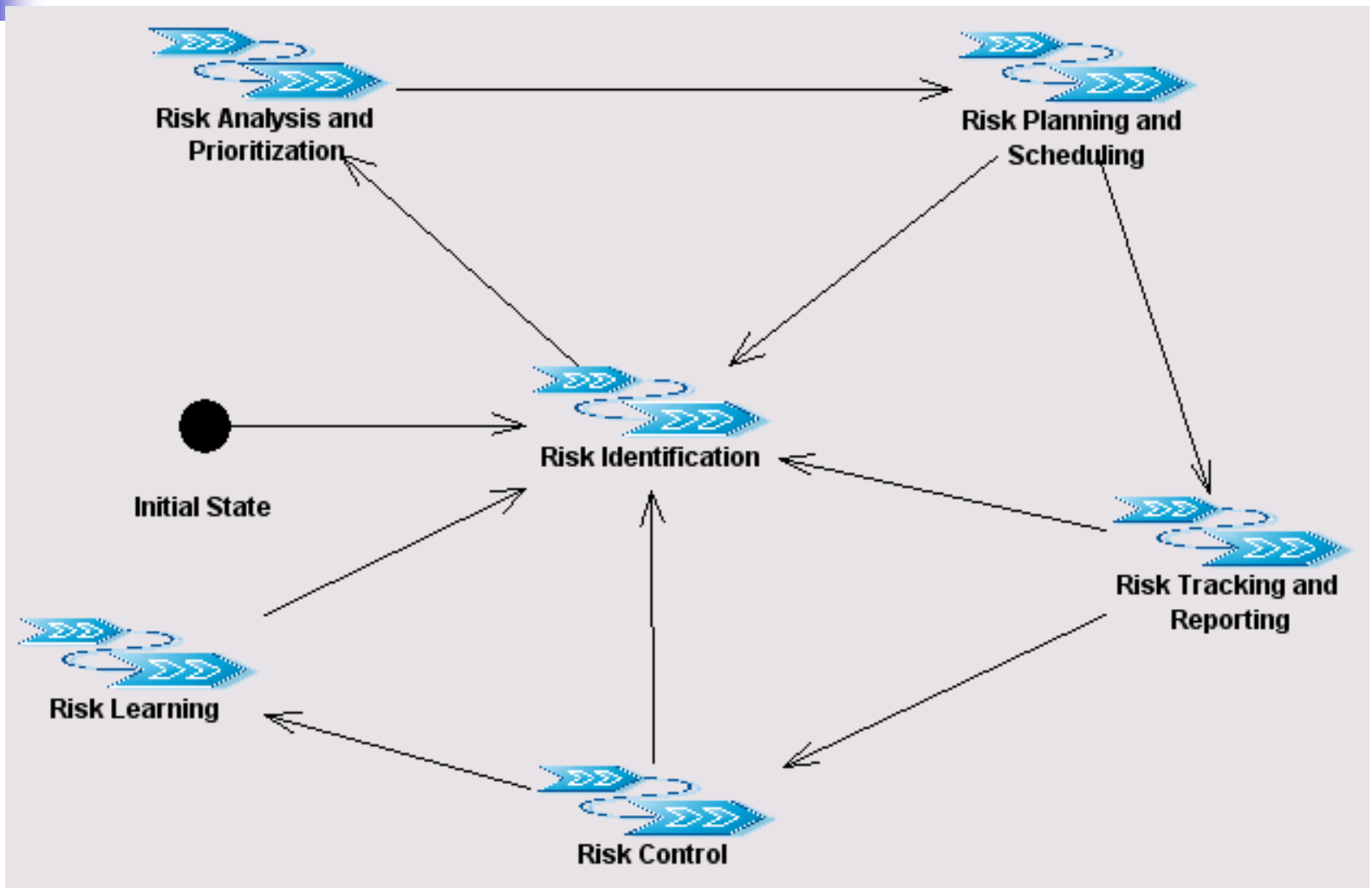# MSF: Baseline Early, Freeze Late
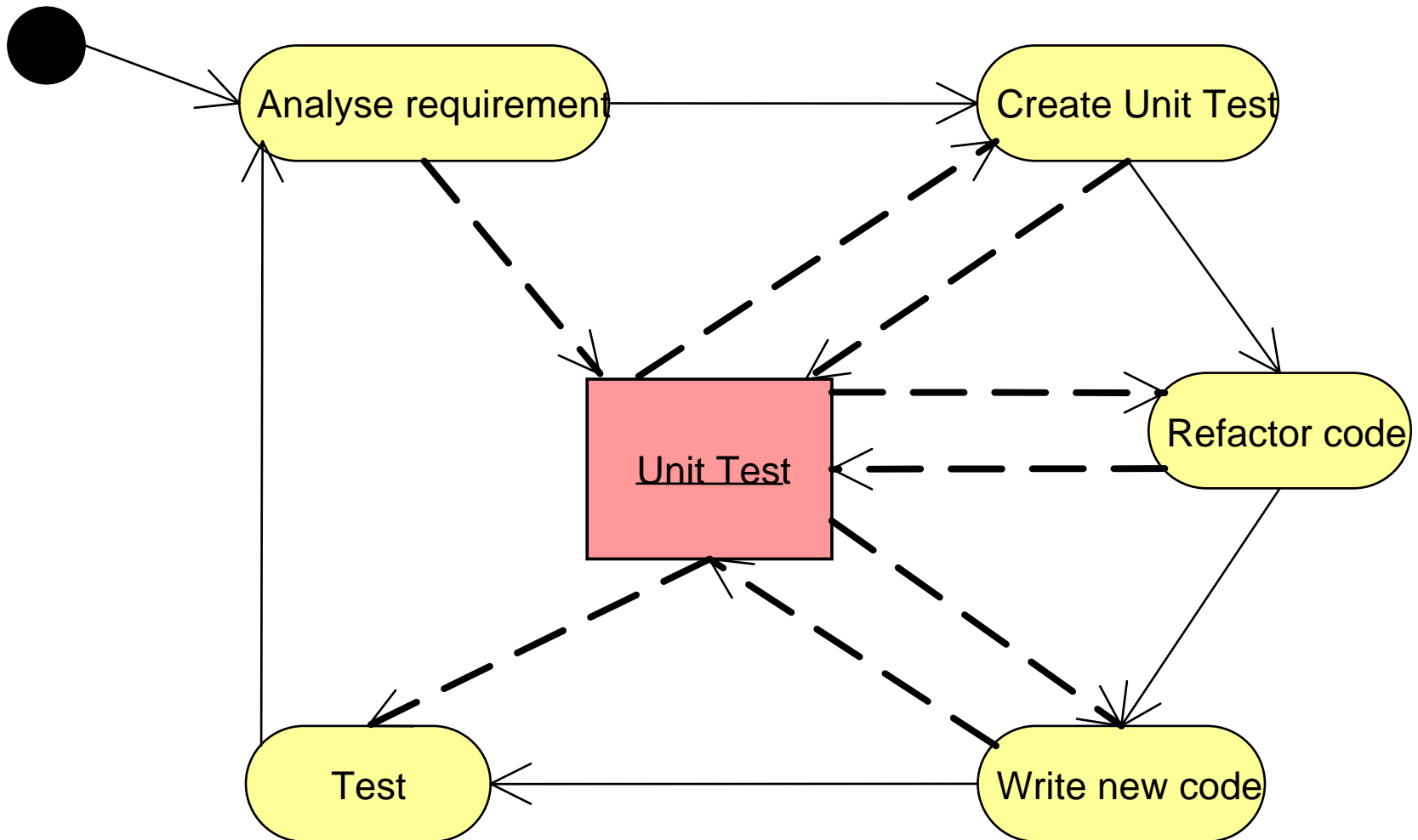
52

Example from XP

Analyse requirement → Create Unit Test

Unit Test
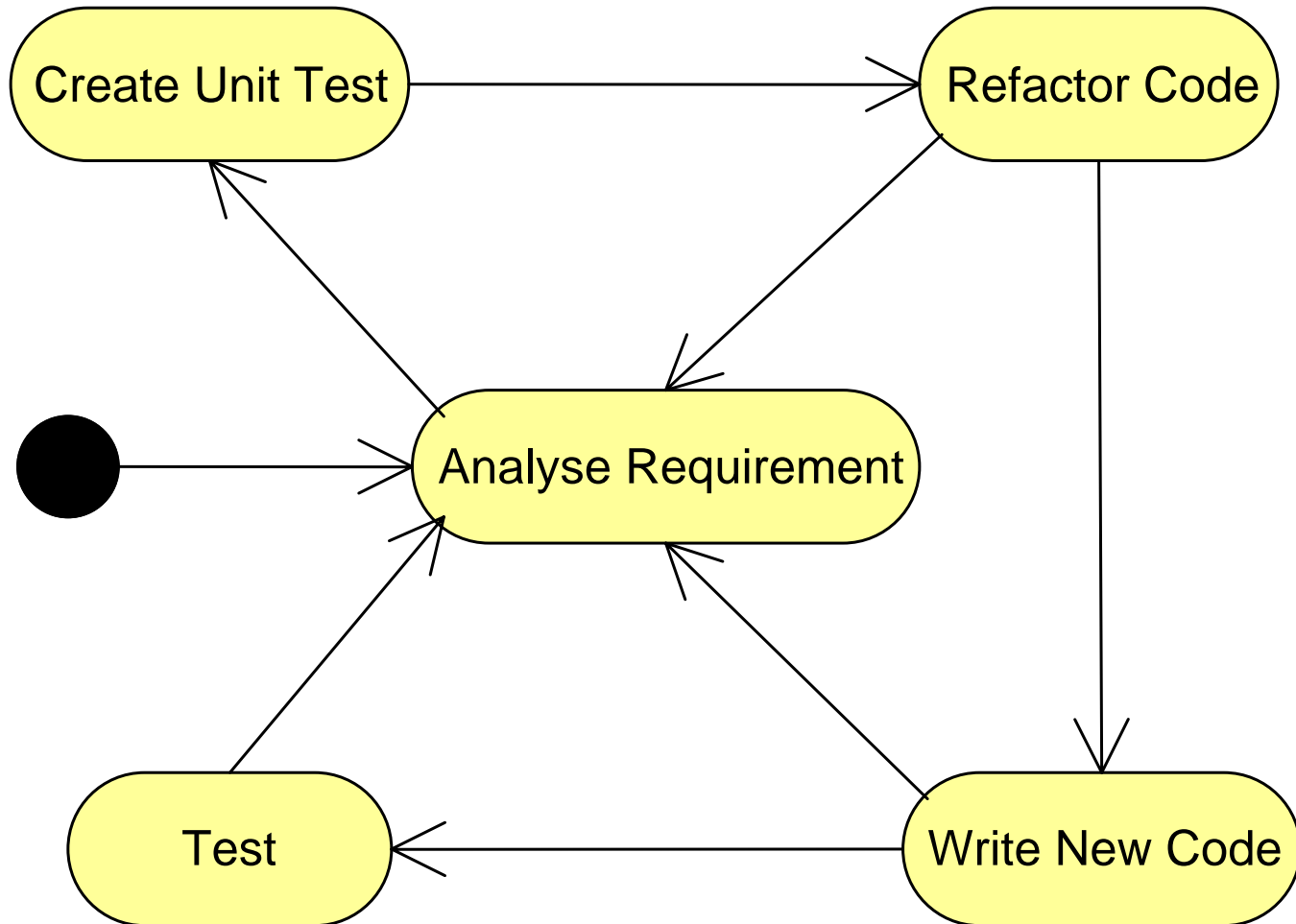
Refactor code

Test

Write new code

53

MSF-based process patterns

- Living document
- Reenterable process

- Smart lifecycle
- Stakeholder-oriented organization

MSF Risk Management Discipline
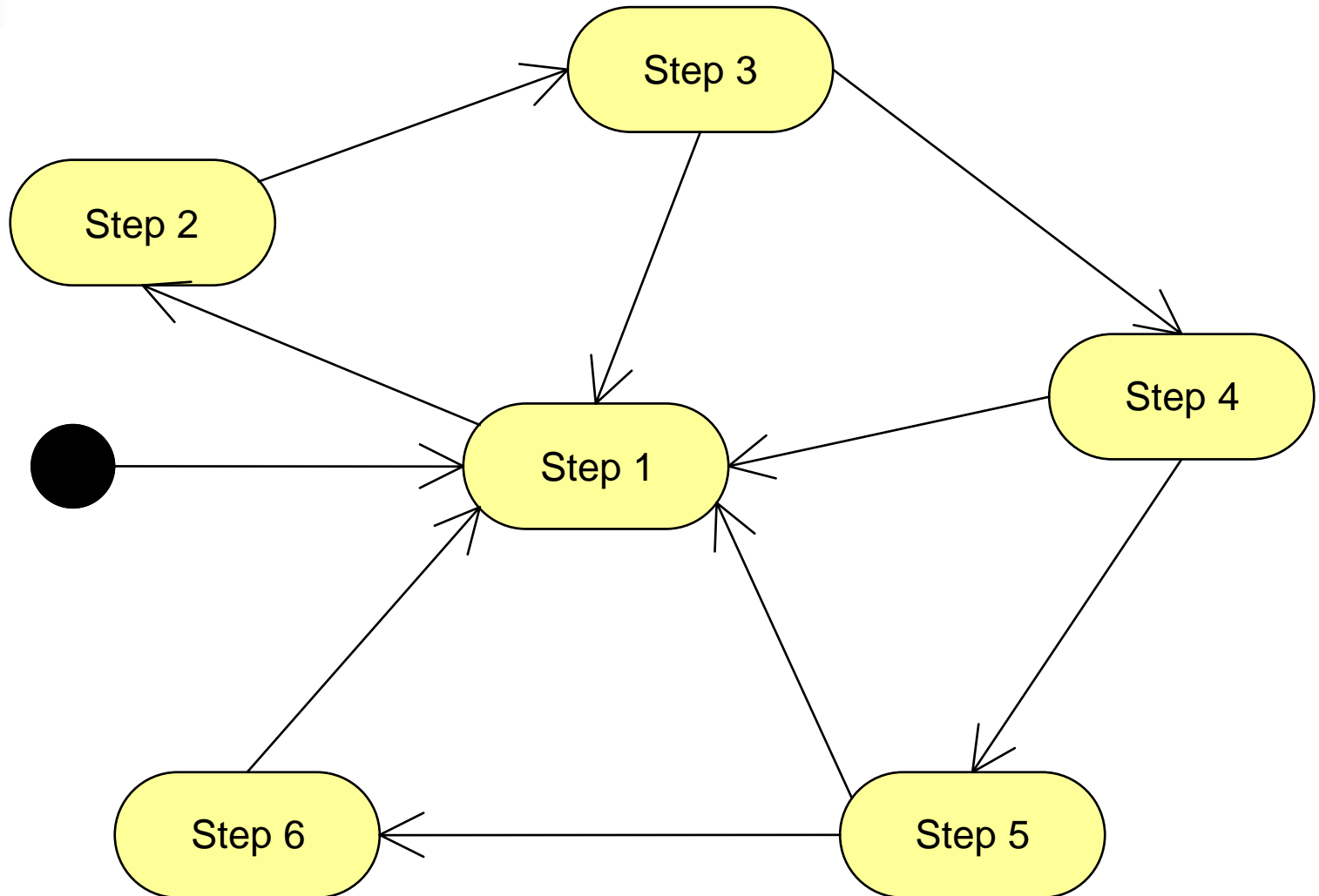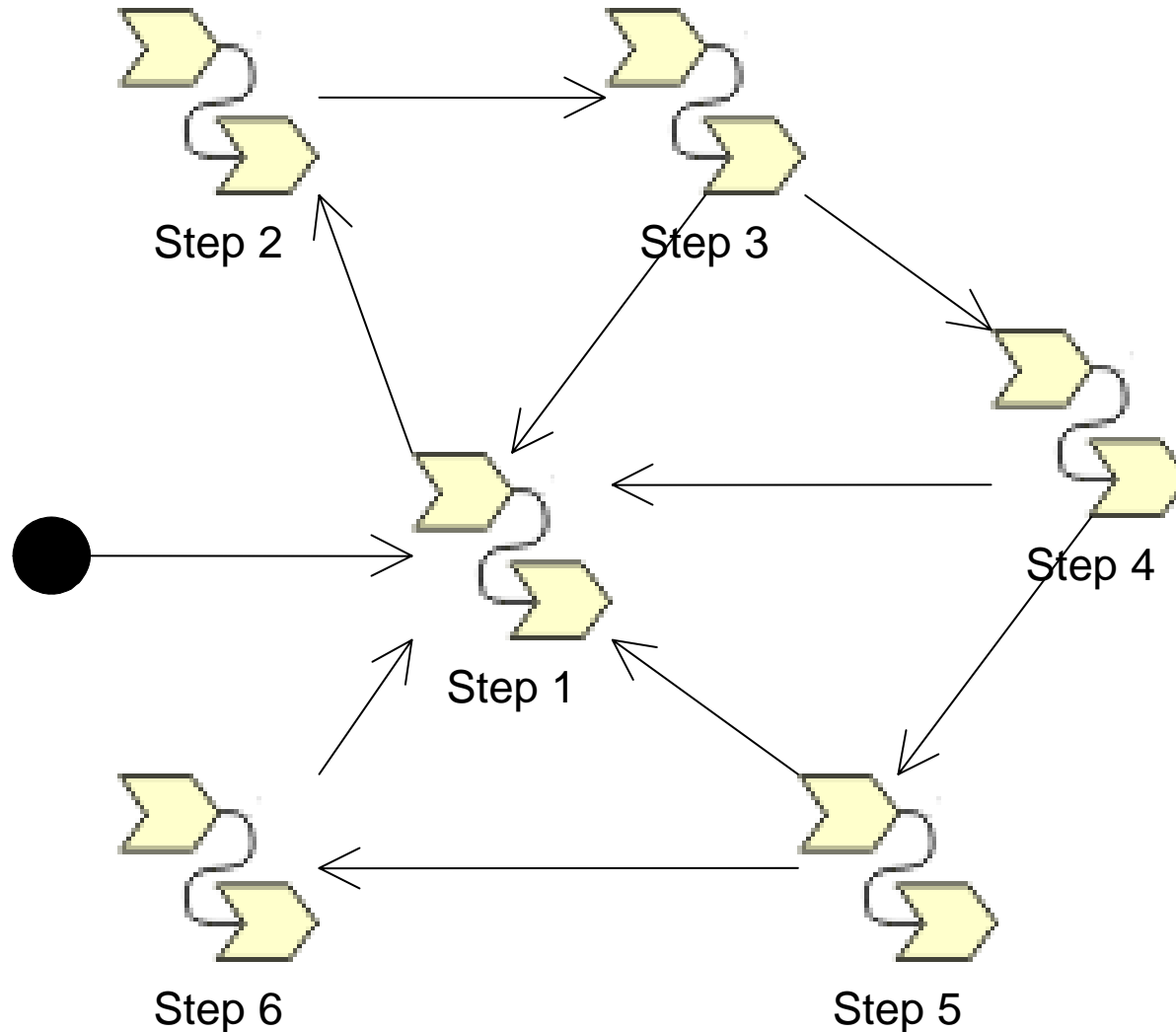
XP Development Cycle

Reenterable process (UML)

Reenterable process (SPEM)

Step 2

Step 3

Step 4

Step 1

Step 6

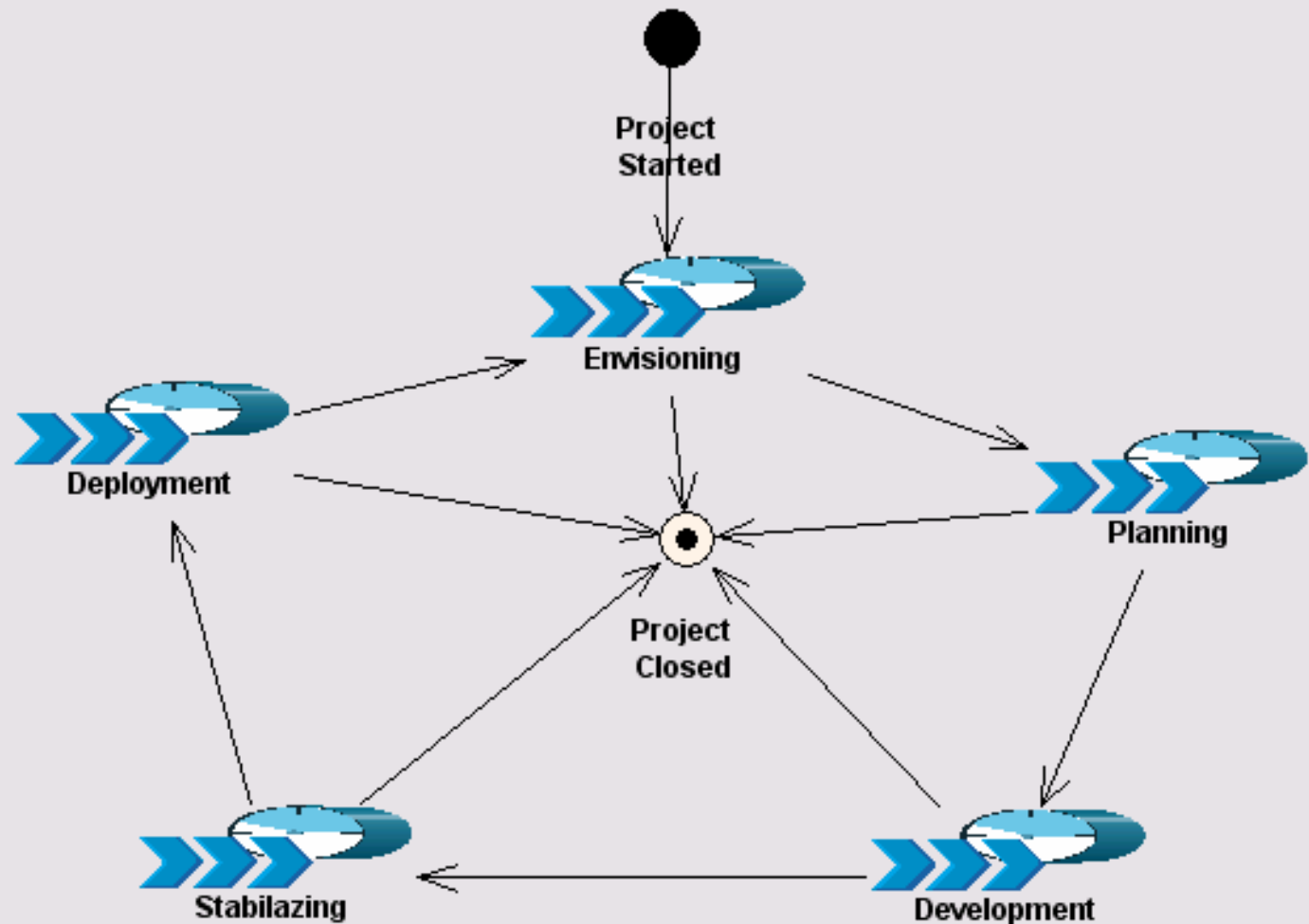Step 5

58

Reenterable process

- ## Intent
  - To provide explicit parallelism for process that deals with set of similar artifacts

- ## Motivation
  - When process should be carried out for a set of similar artifacts (usually we have word "list" in description of artifact) not all of them can be ready to transition to next step. It is reasonable to proceed with each artifact separately but making sure all artifacts go through all process steps
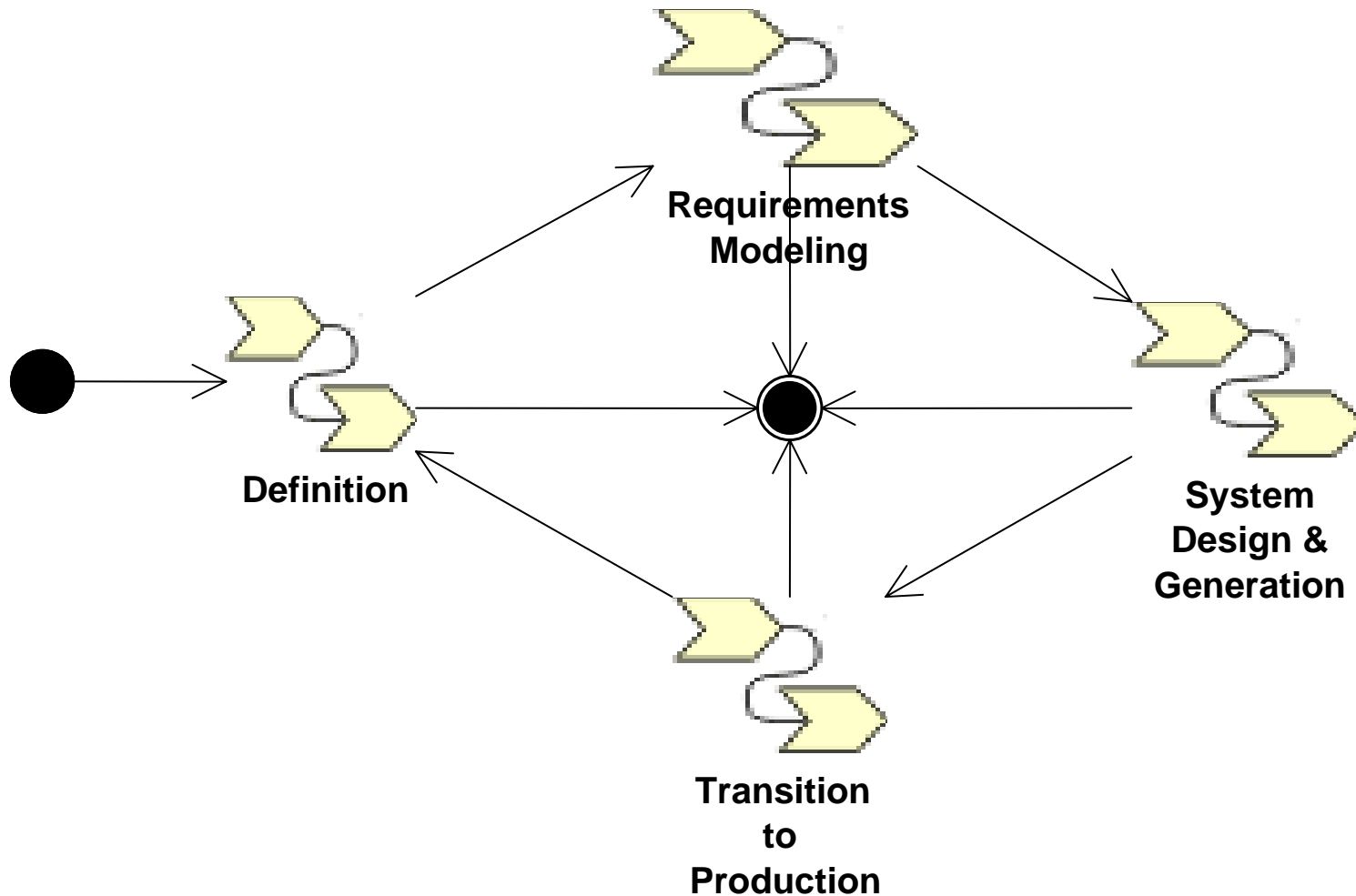
MSF-based process patterns

- Living document
- Reenterable process
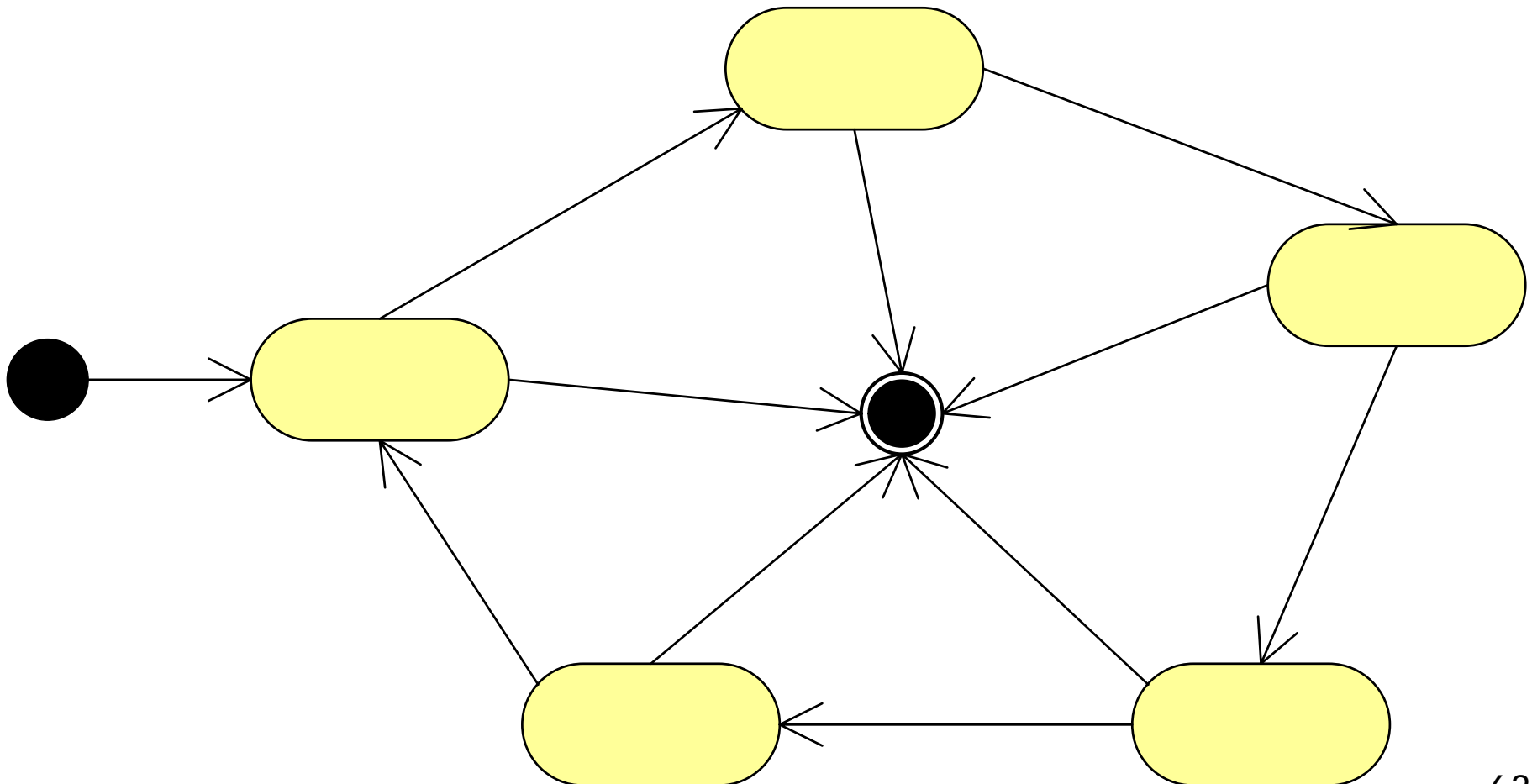- Smart lifecycle

- Stakeholder-oriented organization

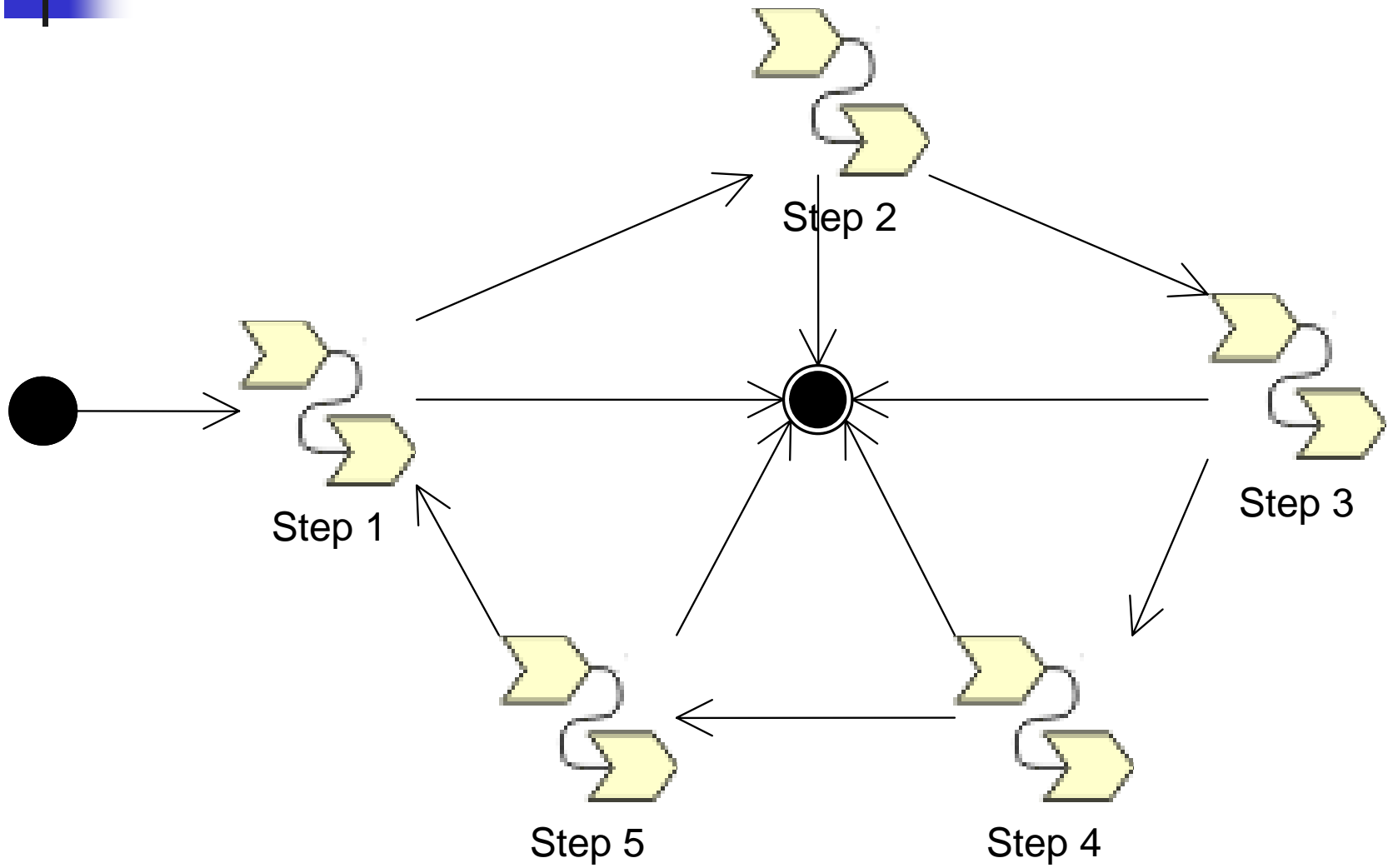# MSF Process Model

61

ORACLE CDM Fast Track

Requirements Modeling

Definition

System Design & Generation

Transition to Production

Smart Lifecycle (UML)

63

Smart lifecycle (SPEM)

Step 1

Step 2

Step 3

Step 4

Step 5

Smart lifecycle

- Intent
  - To provide a possibility of making critical (go/no-go) decisions throughout entire project lifecycle (at the end of each phase)
- Motivation
  - Inability to make an important decision that influences whole project can lead to significant loss of money and time. Closing project at proper time can prevent from further loss of resources invested into project
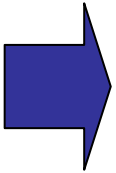
Rationale

- If project has started it means some resources were invested in it

- We may close the project to prevent further loss of money and time

- Most often if the project is not going to finish it is closed after first phases
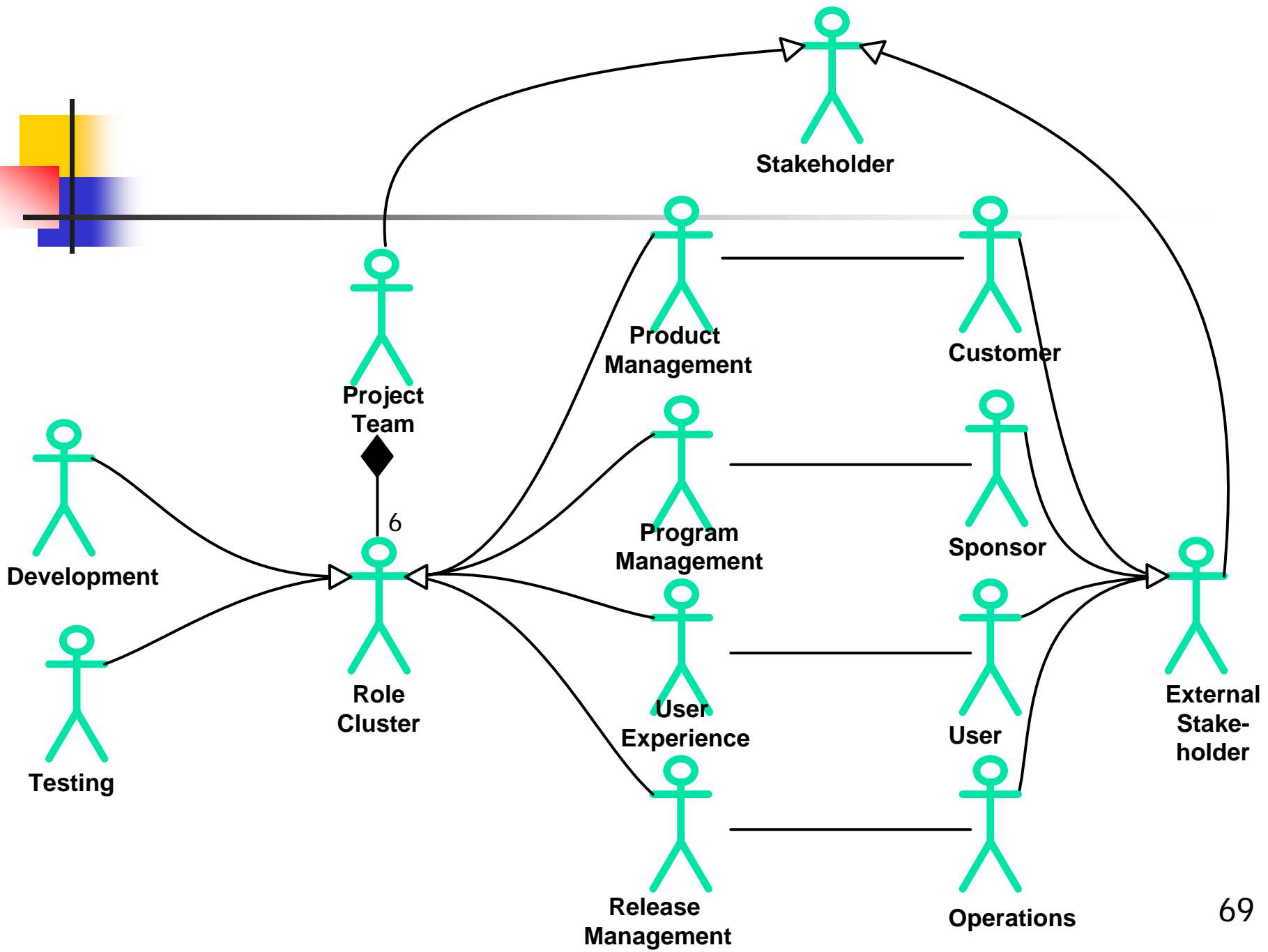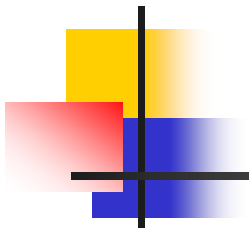
MSF-based process patterns

- Living document
- Reenterable process
- Smart lifecycle
- Stakeholder-oriented organization
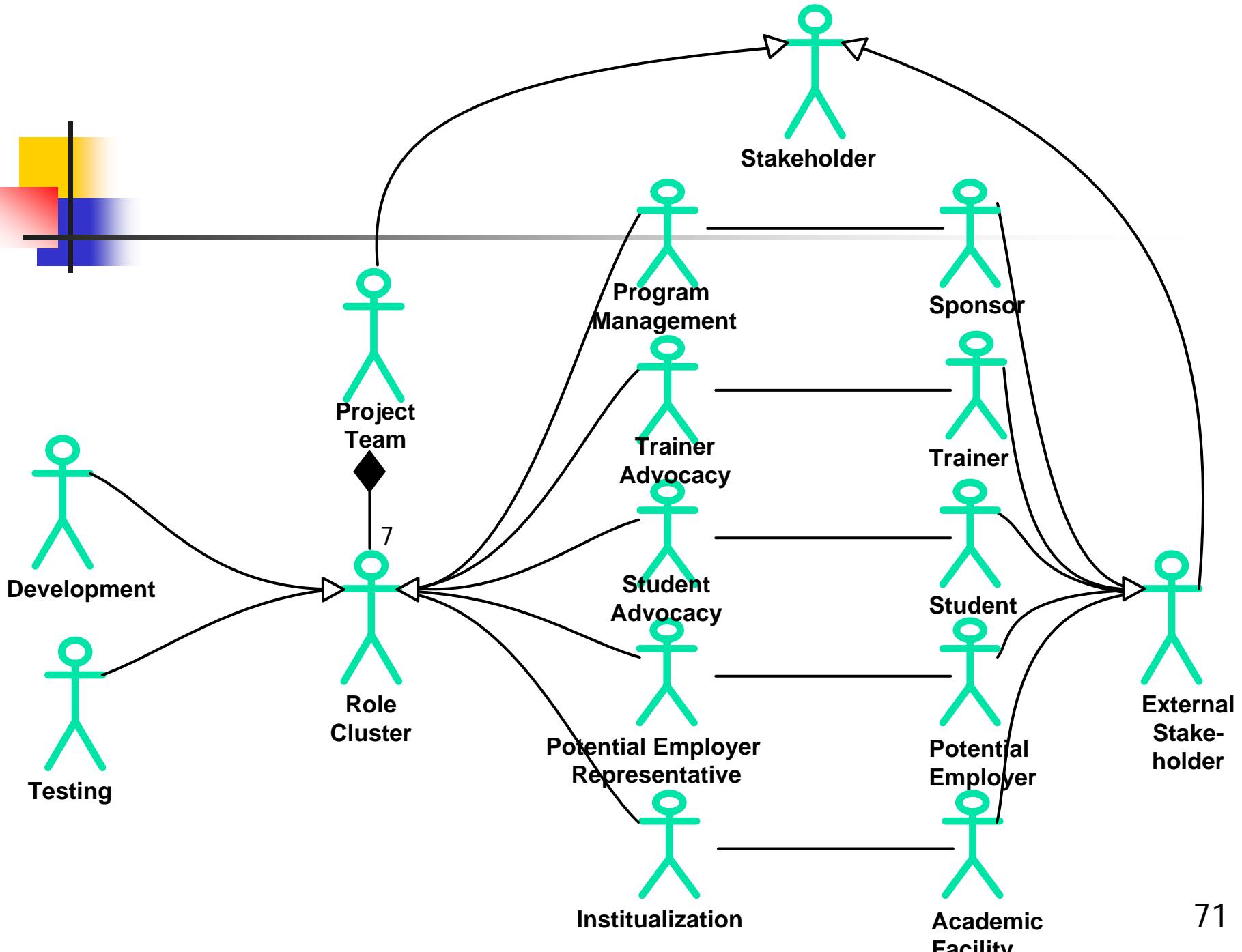
Stakeholders in MSF

- "Classic" MSF has 6 role clusters
  - Product Management
  - Program Management
  - Development
  - Testing
  - Release Management
  - User Experience

Stakeholder

Project
Team

Development

Testing

Role
Cluster

6

Product
Management

Program
Management

User
Experience

Release
Management

Customer

Sponsor

User

Operations

External
Stake-
holder

69

How does it work?

- We used "customized" MSF for courseware development
- We had 7 role clusters
  - Program Management
  - Trainer Advocacy
  - Student Advocacy
  - Potential Employer Representative
  - Institutialization
  - Development
  - Testing

Stakeholder

Program Management

Sponsor

Project Team

Trainer Advocacy

Trainer

Development

Student Advocacy

Student

Testing

Role Cluster

7

External Stake-holder

Potential Employer Representative

Potential Employer

Instiualization

Academic Facility

71

Stakeholder-oriented organization

Stakeholder

Project Team

Role 1

External Stakeholder 1

Role 2

External Stakeholder 2

Role Y

Role

Role X

Role N

External Stakeholder N

External Stakeholder

72

Stakeholder-oriented organization

- Intent
  - To make sure interests of key project stakeholders are taken into account and balanced
- Motivation
  - Naturally different stakeholders have different and often conflicting interests in project and its result. Such organization makes probability of satisfaction of key stakeholders much higher

Examples from XP

- Business people and developers must work together daily throughout the project
- The sponsors, developers, and users should be able to maintain a constant pace indefinitely

Conclusions

- High-level formal description of software process leads to discovery of some process patterns

- Using SPEM allowed to give Gamma-style definitions to discovered patterns
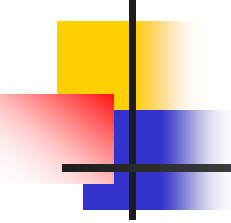
Our thanks to:

- **Nikita Boyko** (Dnepropetrovsk National University, Ukraine)
- **Alex Dubinsky** (Dnepropetrovsk National University, Ukraine)
- **Andrey Terekhov** (Microsoft, Russia)
- **Alex Zverintsev** (eLine Software, Inc., Ukraine/USA)
- **Yevgen Berlyand** (Information Systems Development, Ukraine)
- **Konstantin Runduev** (Dnepropetrovsk National University, Ukraine)
- **Stanislav Petrovsky** (ABV Technique, Ukraine)

Our thanks to:

- **Stanislav Busygin** (University of Florida, USA)
- **Mariele Hagen**
  (University of Leipzig, Germany)
- **Vivienne Suen**
  (Osellus, Inc., Canada)
- **ZEN-Process-Pattern-Team** - M. Gnatz, F. Marschall, G. Popp, A. Rausch, M. Rodenberg-Ruiz, W. Schwerin, K. Bergner (University of Munich, Germany)
- **Stanislav Vonog**
  (Moscow Institute of Physics and Technology, Russia)

- This presentation was delivered on March 4, 2004 in Moscow State University (Russia) at the conference "Microsoft Technologies in Computer Science and Software Engineering"

- You can download this presentation from:
- http://www.vlpavlov.org
  http://www.microsoft.cs.msu.su/conference
  http://www.elinesoftware.com

- Questions?