



Global Information Systems:

Framework (2)

Prof. Dr. Jan M. Pawlowski
03.11.2009



1934 2009

UNIVERSITY OF JYVÄSKYLÄ



Contents

Introduction

A Framework for Global Information Systems

- Process Framework
- Project Lifecycle
- Dimensions

Discussion



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

Assumptions

Scenario: Global Software Development

- Multiple developers in different locations
- Developing software for various markets
- Distributed development, distributed distribution

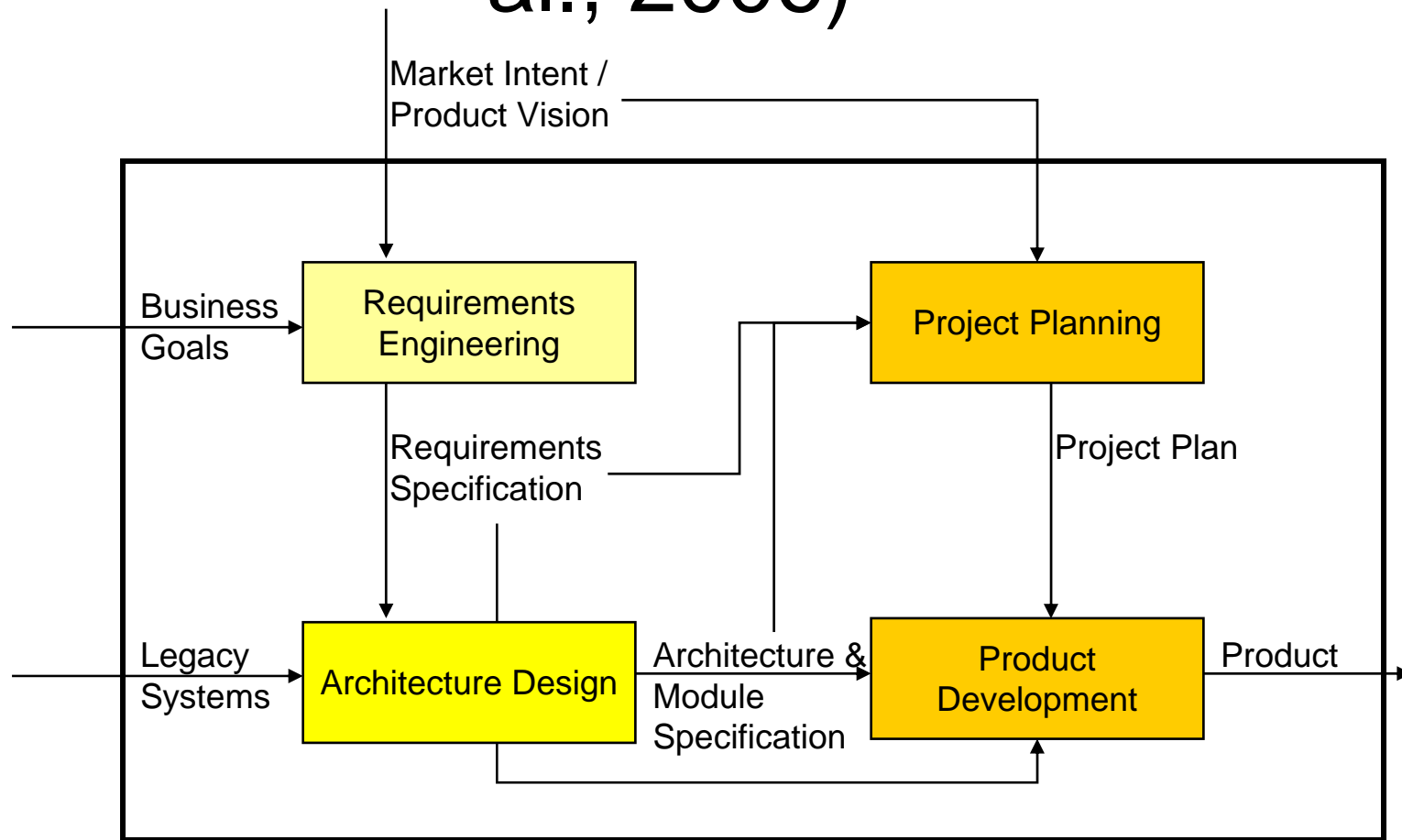
Process Framework

- Detailed discussion of process parts

Assumption: Usage of development models



Process Framework (Sangwan et al., 2006)



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

Decision points (Sangwan et al., 2006)

1. Initiate research
 - Developing new products / services
2. Initiate requirements definition and architecture design
3. Developing a product / service
 - Scope
 - Schedule
 - Investments
4. Releasing a product / service
5. Removing a product / service



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

Process Framework: The Open Unified Process

- Basis to structure the development of global information systems
- Framework for
- Adaptable for
 - E.g., a
 - plugins
- Goals (Eclipse)
 - Collaborate to
 - understand
 - Balance competing priorities to maximize stakeholder value
 - Focus on the architecture early to minimize risks and organize development.
 - Evolve to continuously obtain feedback and improve
- <http://www.eclipse.org/epf>

Remark: This is not a software engineering course, the framework is only used to structure processes and activities! Any other methodology could be used!



The Open Unified Process – Project Lifecycle

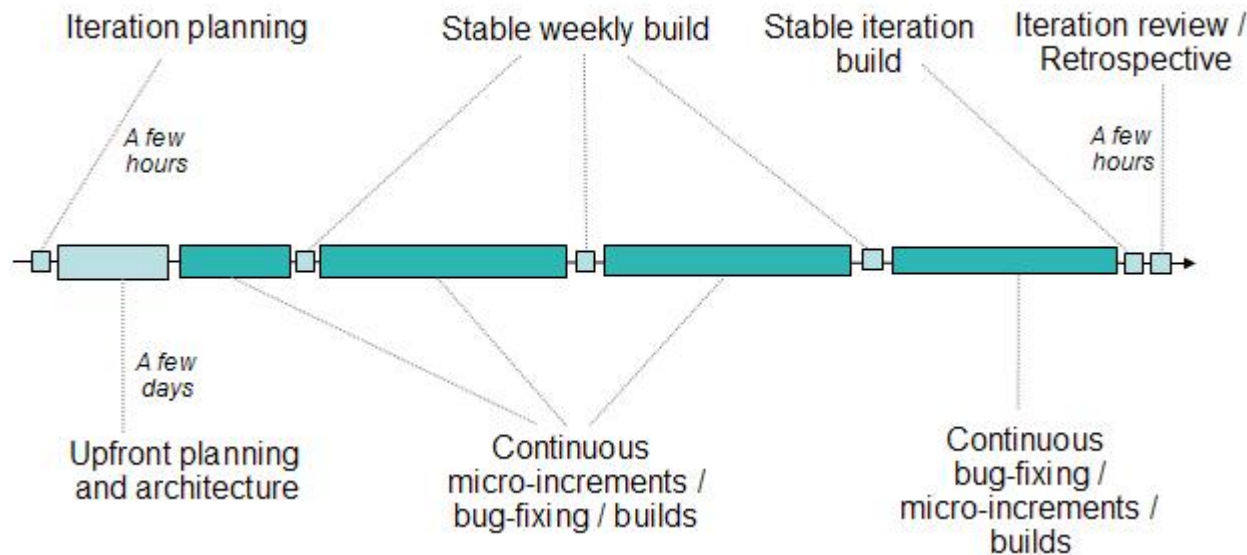
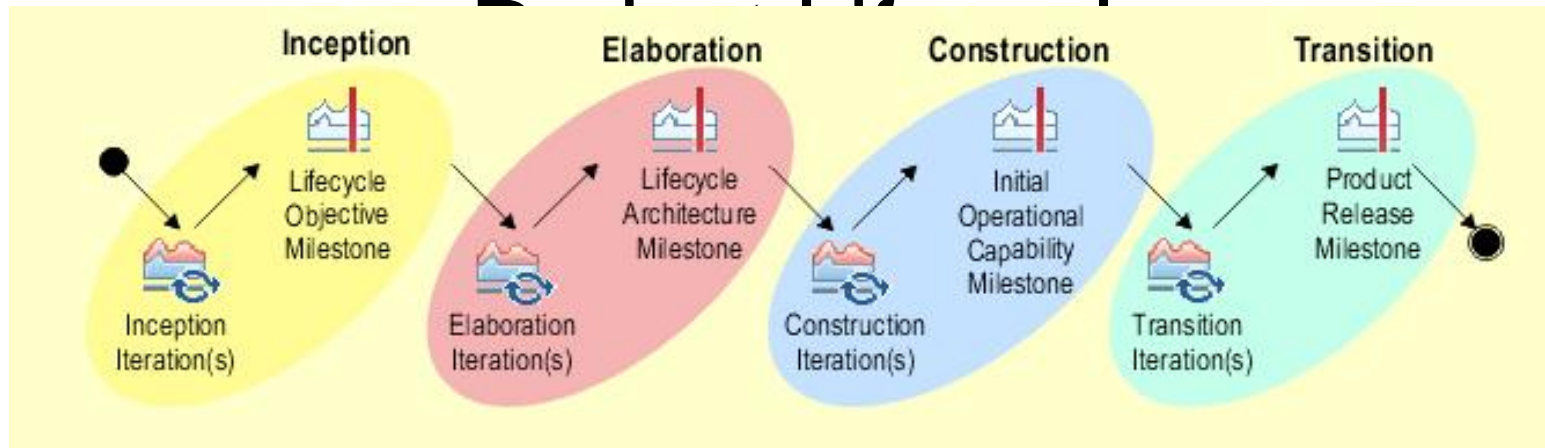
- ❏ Inception. Do we agree on project scope and objectives, and whether or not the project should proceed?
- ❏ Elaboration. Do we agree on the executable architecture to be used for developing the application and do we find that the value delivered so far and the remaining risk is acceptable?
- ❏ Construction. Do we find that we have an application that is sufficiently close to being released that we should switch the primary focus of the team to tuning, polishing and ensuring successful deployment?
- ❏ Transition. Is the application ready to release?



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

The Open Unified Process –



[Source: <http://www.epfwiki.net/wikis/openup/>]



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

The Open Unified Process – Disciplines

- Architecture
- Configuration and Change Management
- Development
- Project Management
- Requirements
- Test

[Source: <http://www.epfwiki.net/wikis/openup/>]



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

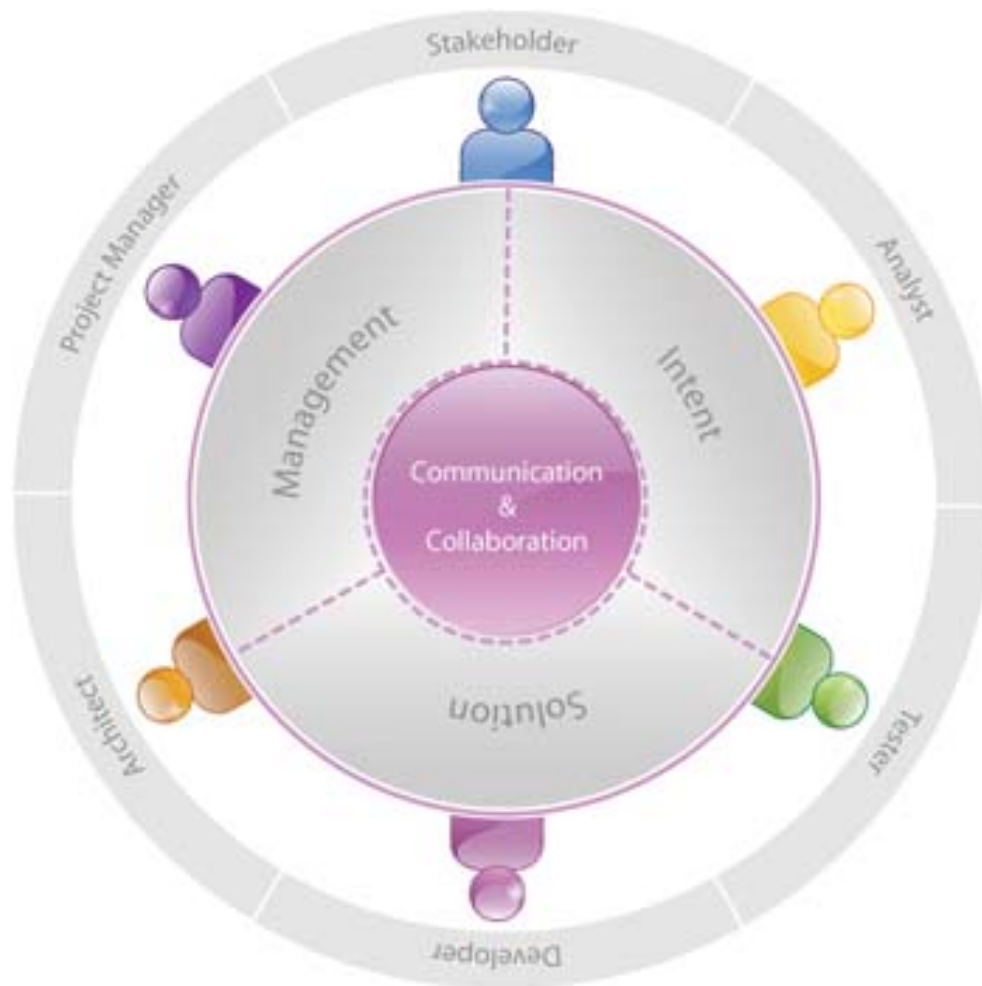
The Open Unified Process – Disciplines

- Structured as *tasks*
- Leading to *work products*
- Architecture
 - Architecture Notebook
- Configuration and Change Management
- Development
 - Design
 - Build
 - Developer Test
 - Implementation
- Project Management
 - Iteration Plan
 - Project Plan
 - Work Items List
 - Risk List
- Requirements
 - Supporting Requirements Specification
 - Vision
 - Use Case
 - Glossary
 - Use-Case Model
- Test
 - Test Case
 - Test Log
 - Test Script

[Source: <http://www.epfwiki.net/wikis/openup/>]



The Open Unified Process – Roles



[Source: <http://www.epfwiki.net/wikis/openup/>]



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

Eclipse Process Framework

The screenshot displays the Eclipse Process Framework Composer interface. The main window shows a tree view of the 'openup_lifecycle' process, which is a Delivery Process (index 0). It contains several iterations and milestones:

- OpenUP lifecycle** (0): Delivery Process
- Inception Iteration [1..n]** (1): Activity
 - Initiate Project* (2): Activity (extends 'initiate_pro...')
 - Plan and Manage Iterati* (5): Activity (extends 'plan_mana...')
 - Identify and Refine Req* (9): Activity (extends 'identify_an...')
 - Agree on the Technical* (13): Activity (extends 'agree_tech...')
- Lifecycle Objectives Mileston** (15): Milestone
- Elaboration Iteration [1..n]** (16): Activity
 - Plan and Manage Iterati* (17): Activity (extends 'plan_mana...')
 - Identify and Refine Req* (21): Activity (extends 'identify_an...')
 - Develop the Architecture* (25): Activity (extends 'develop_ar...')
 - Develop Solution Increm* (34): Activity (extends 'develop_so...')
 - Test Solution* (40): Activity (extends 'test_solutio...')
 - Ongoing Tasks* (44): Activity (extends 'ongoing_ta...')
- Lifecycle Architecture Milestc** (46): Milestone
- Construction Iteration [1..n]** (47): Activity
 - Plan and Manage Iterati* (48): Activity (extends 'plan_mana...')
 - Identify and Refine Req* (52): Activity

The 'Properties' window shows the configuration for the selected activity, **Activity: manage_iteration**.

General Information

Provide general information about this Activity.

Name: manage_iteration

Presentation name: Plan and Manage Iteration

Options:

- Optional
- Multiple Occurrences
- Planned
- Suppress
- Event Driven
- Ongoing
- Repeatable

Table:

Index	Presentation Name	Dependency
-------	-------------------	------------

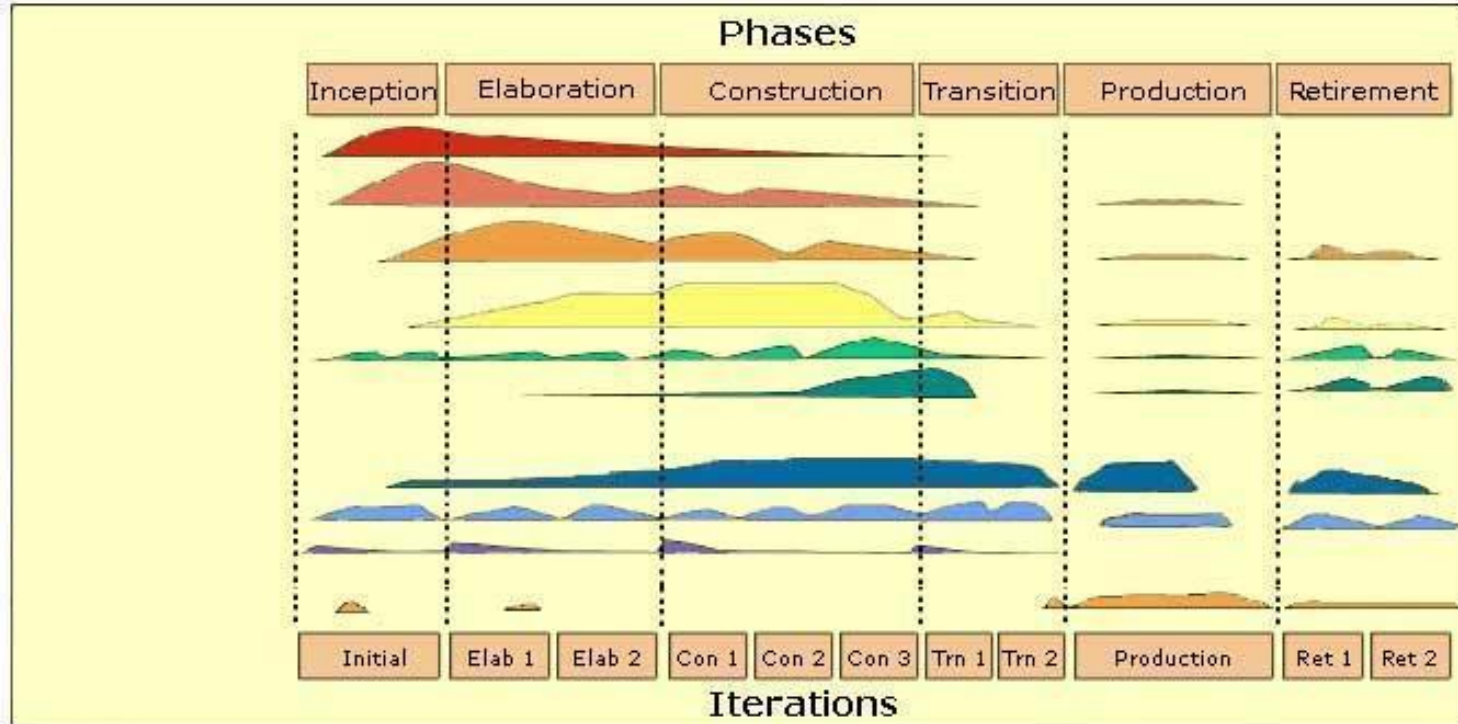
Extensions: Enterprise Unified

Development Disciplines

- Business Modeling
- Requirements
- Analysis & Design
- Implementation
- Test
- Deployment

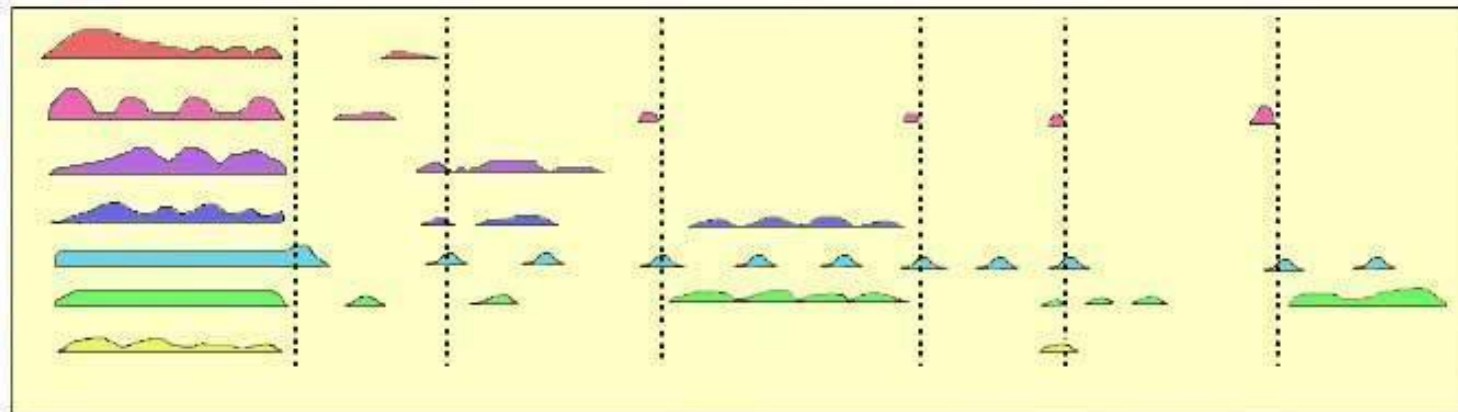
Support Disciplines

- Configuration and Change Mgmt.
- Project Management
- Environment
- Operations & Support



Enterprise Disciplines

- Enterprise Business Modeling
- Portfolio Management
- Enterprise Architecture
- Strategic Reuse
- People Management
- Enterprise Administration
- Software Process Improvement



Copyright 2003-2005 Scott W. Ambler

Summary

- ❏ Unified Process as a basis for software development process
- ❏ Focus on different aspects of the lifecycle
 - E.g., risk management, communication
- ❏ Extension model for globally distributed processes and stakeholders



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

The Open Unified Process – Disciplines

- Architecture
 - Architecture Notebook
- Configuration and Change Management
- Development
 - Design
 - Build
 - Developer Test
 - Implementation
- Project Management
 - Iteration Plan
 - **Project Plan**
 - Work Items List
 - **Risk List**
- Requirements
 - Supporting Requirements Specification
 - Vision
 - Use Case
 - Glossary
 - Use-Case Model
- Test
 - Test Case
 - Test Log
 - Test Script
- Roles
- Artefacts / Support

[Source: <http://www.epfwiki.net/wikis/openup/>]



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

Project Planning

- Planning of the process
 - Distribution of actors / organization
 - Staff selection
 - Cost estimation
 - Schedule / workflow
 - Coordination activities
 - Communication tools
- Results
 - Project plan, workflow, ...
 - Staff plan: roles / competencies / effort
 - Coordination planning
 - Supporting tools: Guidelines / rules / standards



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

Project Planning: Offshore vs. Nearshore

❏ “Offshoring”

- Assigning (parts of) a knowledge-intensive development process to a geographically remote partner
- Cost reduction
- Accelerating the production process

❏ “Nearshoring”

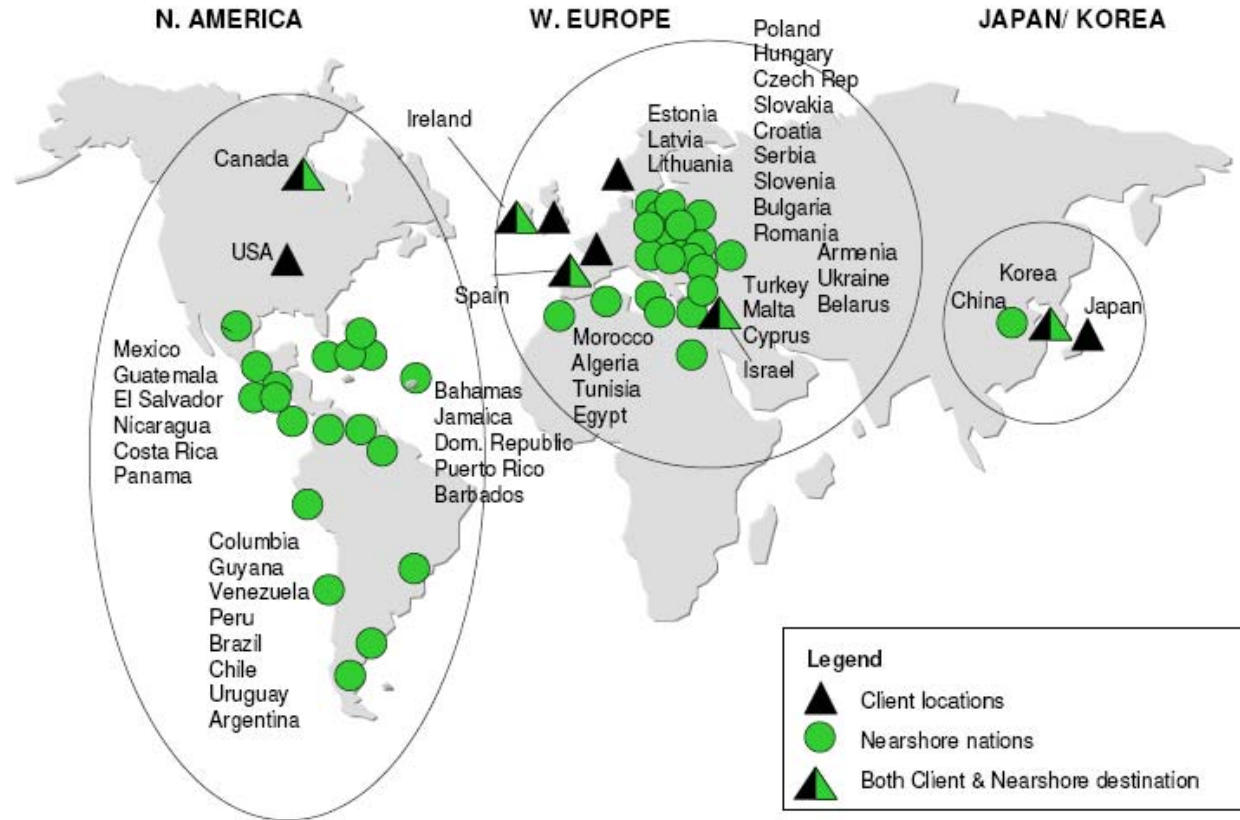
- Assigning (parts of) the development process to a geographically close partner
- Possible advantages concerning distance, language, time, culture, politics, ...



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

Project Planning: Offshore vs. Nearshore (Carmel. Abbott. 2006)



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

Project Planning: Offshore vs. Nearshore

- Need to analyze and estimate...
 - Coordination
 - Cultural misunderstandings
 - Communication cost
 - Team building / training cost
 - Mistakes / prolonged life cycle
- Findings for India (Carmel, Abbott, 2006)
 - Nearshore locations provide a politically stable atmosphere
 - India is a long way away
 - India is called “distant lands;” difficulties with long distance management and cultural differences
 - India is too difficult to manage remotely; too many time zones away. Cheaper, real-time communication relative to India.
 - Nearshore better for outsourcing business-critical work
 - Nearshore offers lower costs of communication, shipping and tariffs



Project Planning: Cost estimation (Sangwan et al., 2006)

- Calibrate cost estimation tool
- Estimate module sizes
- Allocate modules to development iterations
- Estimate code size for each iteration
- Estimate development time, effort and peak staff
 - Including coordination / communication effort
- Estimate iteration development time and average team staff size
- Estimate development schedule time
 - Including time differences
- Estimate development cost



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

Project Planning: Sample coordination activities

Instruments to monitor and control the development process (Boland, Fitzgerald, 2004)

- ❏ Single software manager and weekly task reports
 - Reducing coordination efforts
 - Tools to assign tasks properly
- ❏ Delivery reports
 - Awareness
 - Trust
- ❏ Quarterly synchronisation meetings
- ❏ Informal meetings and instruments



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

Project Planning: Risk Management

- Risk: The possibility of suffering a loss (Sangwan et al, 2006)
- Risk lifecycle
 - Identify
 - Analyze
 - Plan
 - Track
 - Control
 - Communicate
- Risk in GSD processes
 - Coordination
 - Architecture alignment
 - Uncertainty and change



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

Risk Management: Identifying risks

- ❏ Organizational aspects / coordination capability
 - Background and skills
 - Domain knowledge of teams
 - Communication / collaboration history
 - Organizational separation / integration
 - Shared culture / language
- ❏ Organizational stability



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

Risk Management: Avoiding risks

- Monitoring and control
- Skill improvement and training
- Unified tool structure
- Management communication
- Frequent builds / prototypes
- Frequent status meetings
- Cross-team reviews
- Contingency planning: If something goes wrong...



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

At the end of this phase, the following results should be ready:

- Project plan
 - Outsourcing / offshoring decisions and agreements
 - Cost planning
- Adapted process model
 - E.g., Global OpenUP
- Risk management
- Coordination planning (to be refined)



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

Summary

- ❏ Distributed development processes lead to new aspects regarding project planning
- ❏ A variety of decisions before the development process influence project success
 - Distribution of actors
 - Coordination activities
 - Management strategy
- ❏ Cost estimation must include influence factors
 - Not all factors can be estimated sufficiently in advance
 - Supporting instruments have to be taken into account
- ❏ Complex process, decision alternatives should be taken into account



Questions

- ❏ How does the Unified Process support global development processes?
- ❏ Which dimension of the UP have to be extended when working in a global context
- ❏ Which aspects are different in the project planning phase between in-house and off-shore development?
- ❏ How to estimate the costs for a global development project?
- ❏ Which supporting mechanisms can be used to improve coordination?
- ❏ Develop a risk profile for a GSD project.



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

References

- ❏ Carmel, E., Abbott, P. (2006):
Configurations of Global Software
Development: Offshore versus Nearshore,
ICSE 2006.
- ❏ Boland, D., Fitzgerald, B. (2004):
Transitioning from a Co-Located to a
Globally-Distributed Software Development
Team : A Case Study at Analog Devices
Inc., In: GSD Workshop, 2004



1934 2009

UNIVERSITY OF JYVÄSKYLÄ

Contact Information ITRI

Prof. Dr. Jan M. Pawlowski

jan.pawlowski@titu.jyu.fi

Skype: jan_m_pawlowski

Office:

Telephone +358 14 260 2596

Fax +358 14 260 2544

<http://users.jyu.fi/~japawlow>



1934 2009

UNIVERSITY OF JYVÄSKYLÄ