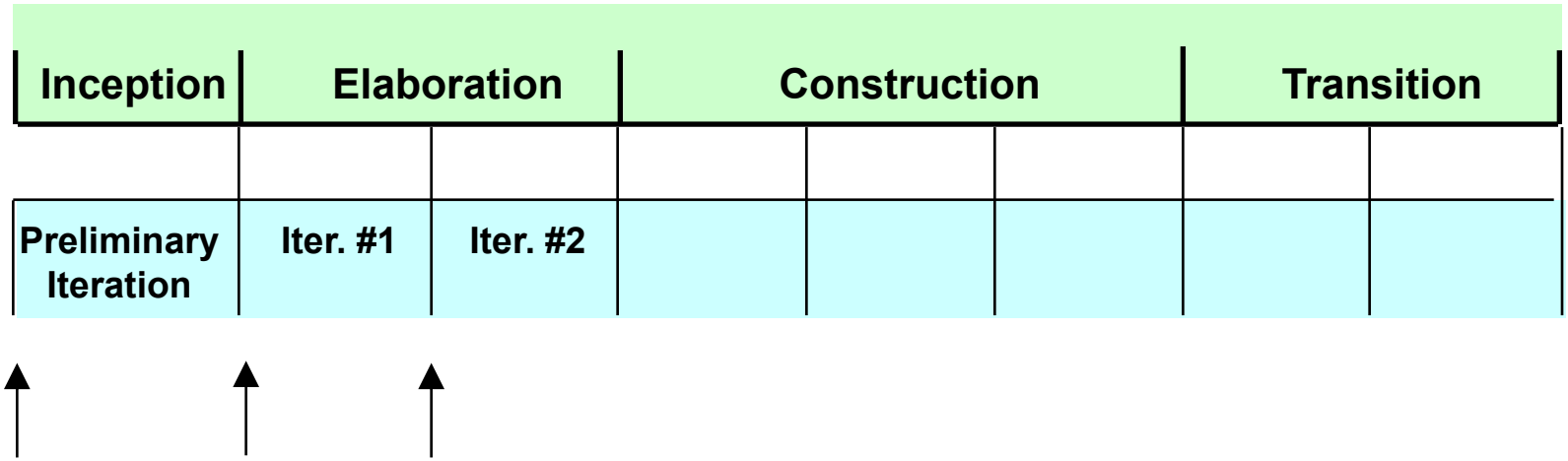
CSC-331
SOFTWARE ENGINEERING

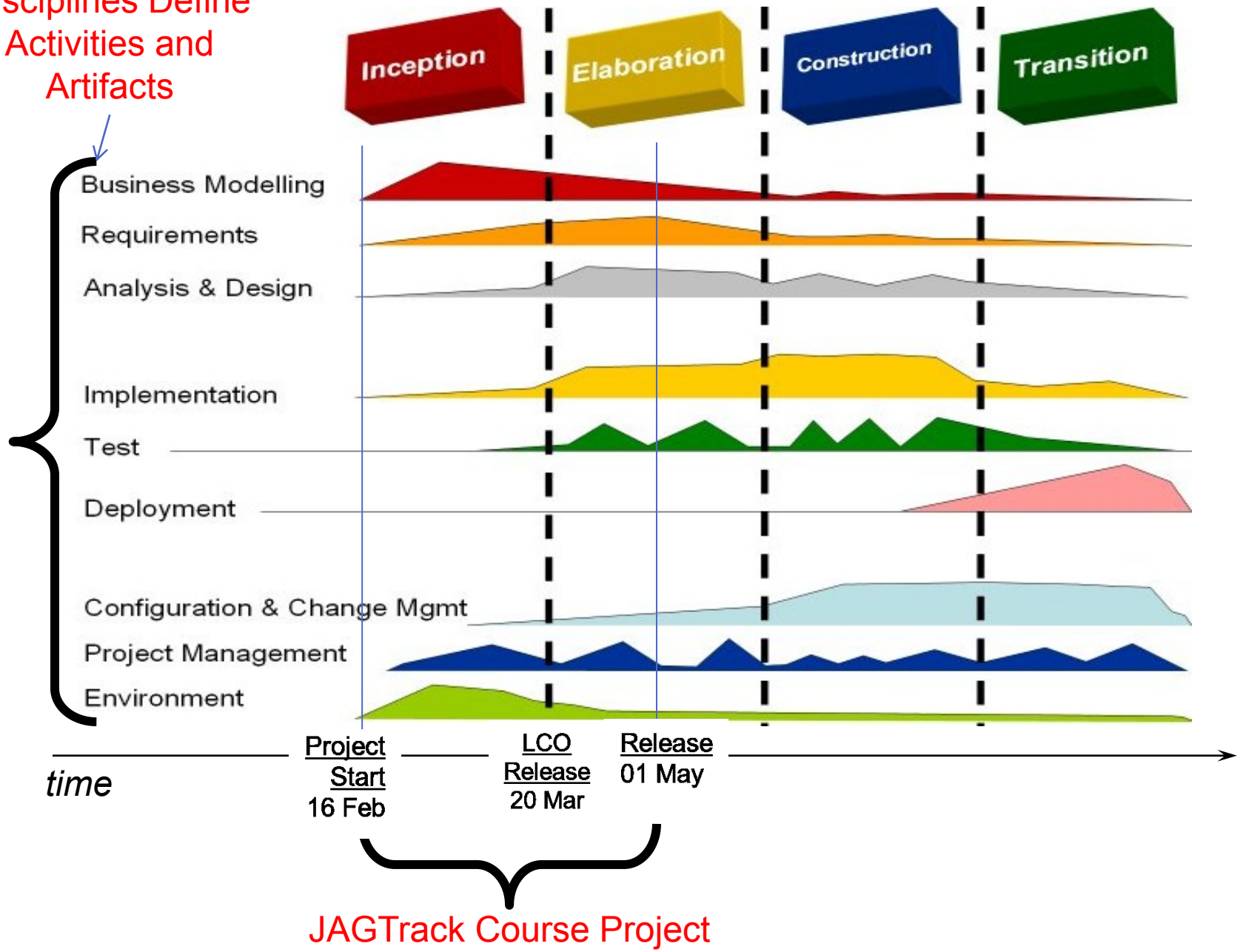
JAGTrack Project

- CSC-331 Project Overview
- RUP Artifacts Template

10	Th	16 Feb	Group Project Kickoff Meeting		
16	**Tu**	**20 Mar**	**Project LCO Briefing**		**Project Briefing/Demo**
17	Th	22 Mar	Group Project Meeting		
18	Tu	27 Mar	Group Project Meeting		
28	**Tu**	**01 May**	**Project Iteration Briefing**		**Project Briefing/Demo**



Disciplines Define Activities and Artifacts

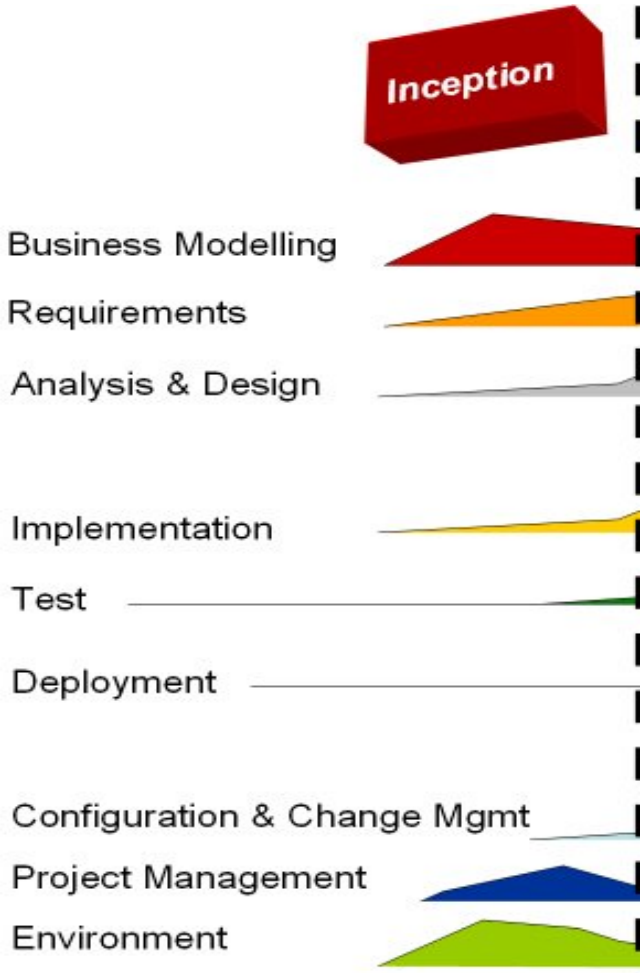


Discipline	Artifacts	Incep I1	Elab E1 .. En	Const C1 .. Cn	Trans T1..T2
Business Modeling	Domain Model	s	r		
	Business rules	s	r		
Requirements	Vision	s	r		
	Use Case model	s	r		
	Supplementary specification	s	r		
	Stakeholder requests	s	r		
	Glossary	s	r		
	Prototypes and proof-of-concept	s	r		
	Requirements management plan	s	r		
	Software requirements specification	s	r		
Analysis & Design	Software architecture document		s		
	Analysis/design model		s		
	Data model/interface specification		s		
Implementation	Integration build plan		s		
Test	Test plan		s		
Deployment	Deployment plan		s		
Configuration Management	Configuration management plan	s	r		
Project Management	Iteration plan	s	r		
	Software development plan (includes Phase plan)	s	r		
	Business case	s	r		
	Risk list	s	r		
	Risk management plan	s	r		
	Quality assurance plan	s	r		
Environment	Development case	s	r		
	Test Guidelines		s		
	Programming Guidelines		s		

Development Case

Inception Artifact Requirements at LCO

Discipline	Artifacts	Incep I1
Business Modeling	Domain Model	s
	Business rules	s
Requirements	Vision	s
	Use Case model	s
	Supplementary specification	s
	Stakeholder requests	s
	Glossary	s
	Prototypes and proof-of-concept	s
	Requirements management plan	s
	Software requirements specification	s
Configuration Management	Configuration management plan	s
Project Management	Iteration plan	s
	Software development plan (includes Phase plan)	s
	Business case	s
	Risk list	s
	Risk management plan	s
	Quality assurance plan	s
Environment	Development case	s



Iteration plan

Domain Model: Class Diagram, Interaction Diagrams

Use Cases (text/diagram), Supplementary Spec

Architectural proof of concept, Feasibility study
User-interface prototype

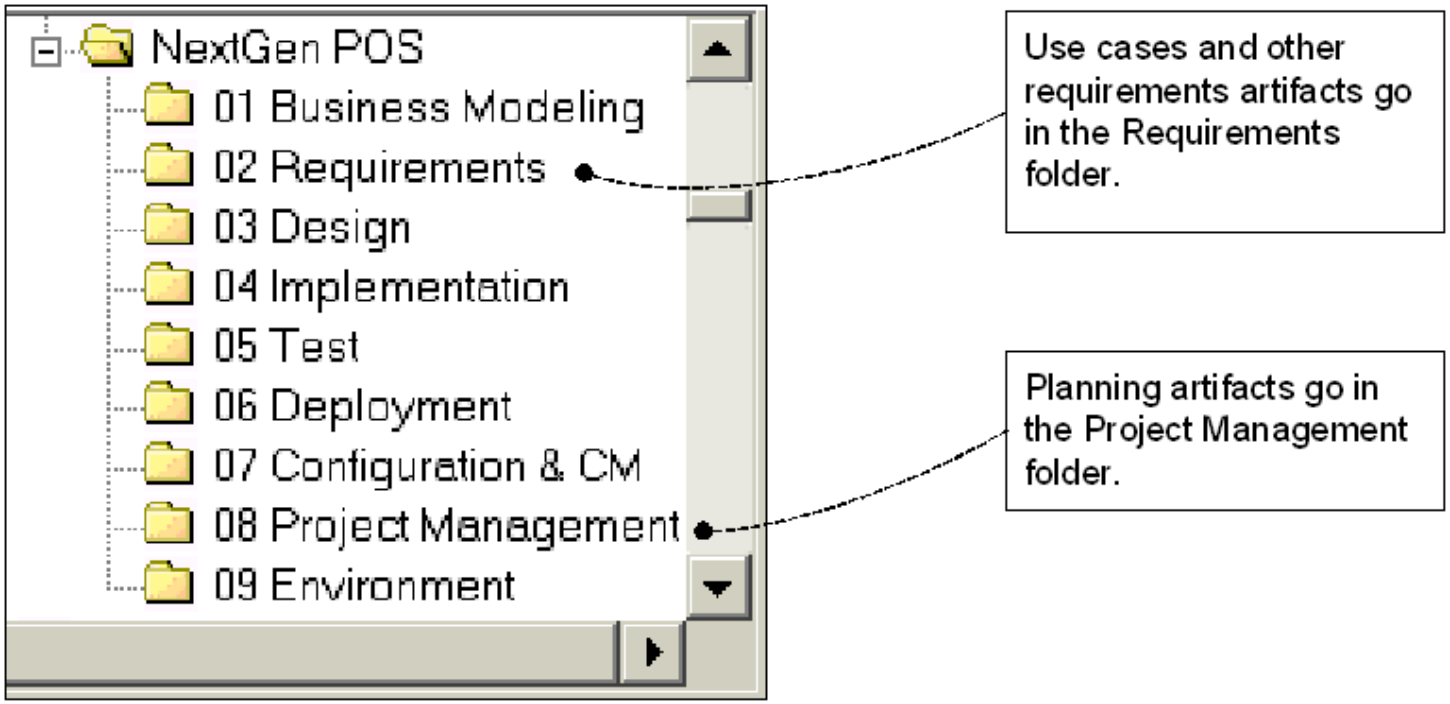
Test management / research (security)

Software development plan, Development case, Vision and business case, Risk management, Glossary, Stakeholder requirements

Environment for development team

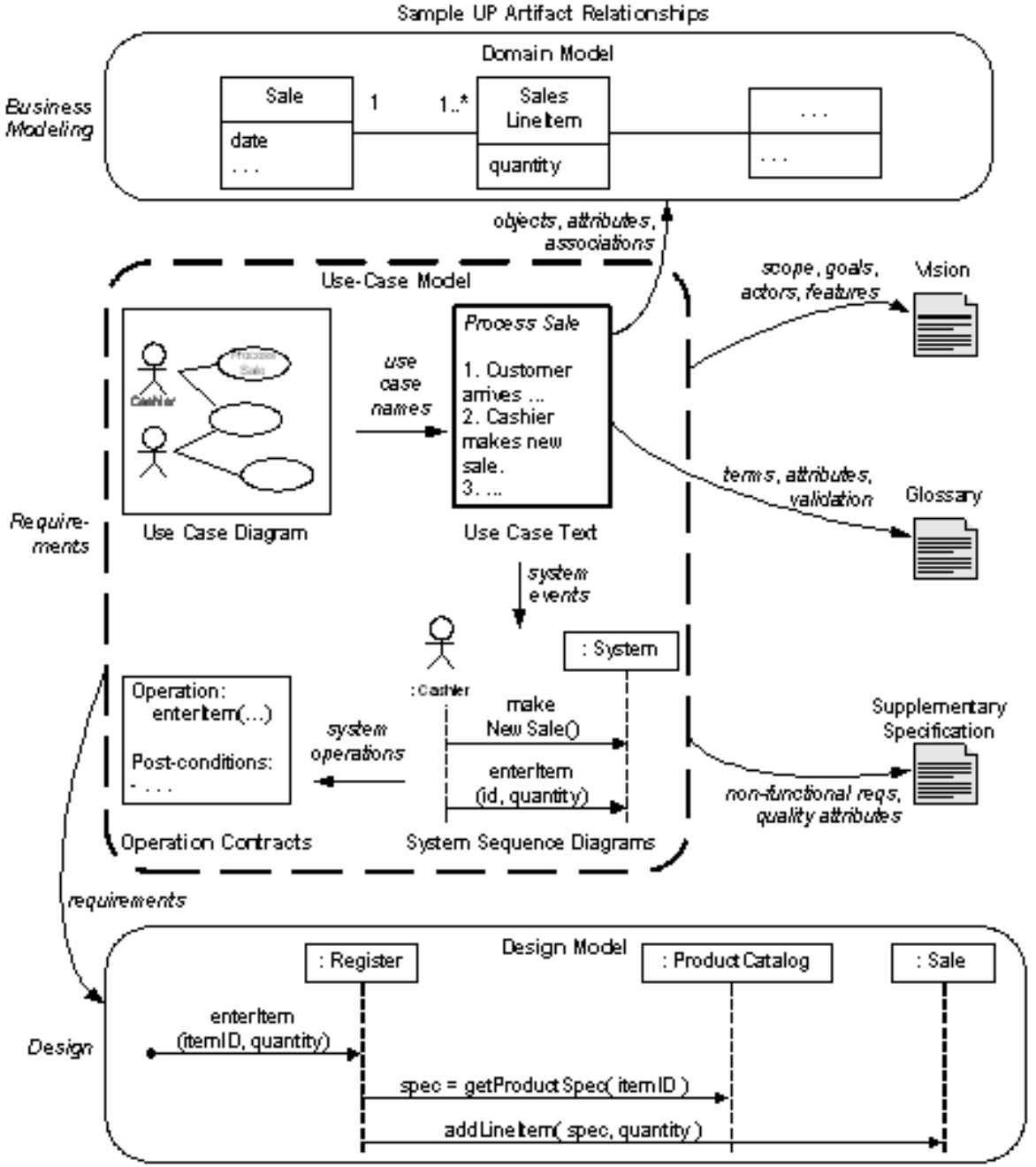
[La05] **Lifecycle Objective** = Concurrence on scope, constraints, worth doing, feasibility (at least one way to do it), tailoring (how we will do it), estimates

- UP organizes artifacts in terms of disciplines…



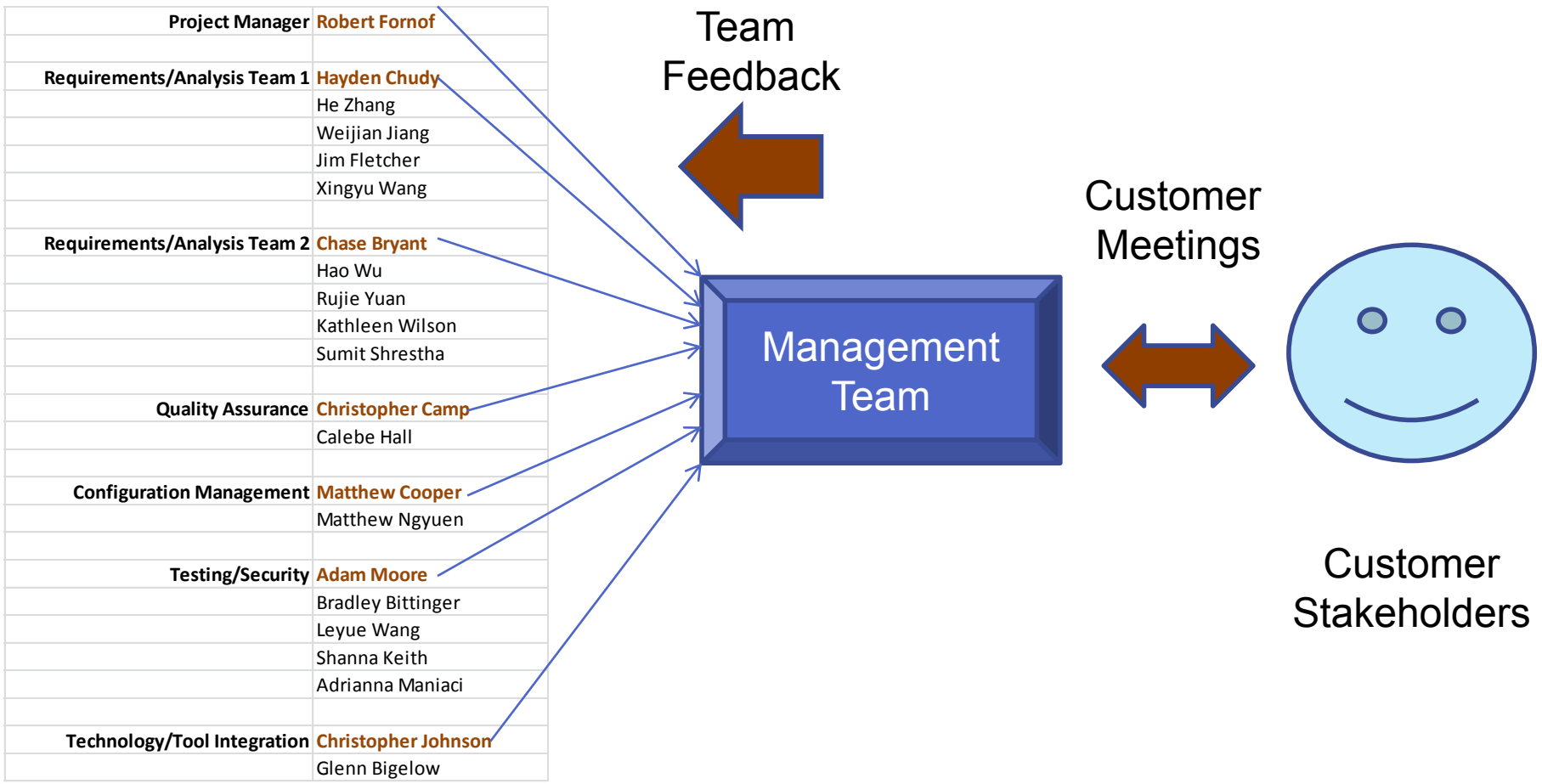
Can be accomplished via a website or wiki

[La05]


Sample UP Artifact Relationships
Business Modeling
Domain Model
Sale
date
. . .
1
1..*
Sales LineItem
quantity
. . .
. . .
objects, attributes, associations
Use-Case Model
scope, goals, actors, features
Vision
Process Sale
1. Customer arrives ...
2. Cashier makes new sale.
3. ...
use case names
Cashier
Use Case Diagram
Use Case Text
terms, attributes, validation
Glossary
Require-ments
system events
: System
: Cashier
Operation:
enterItem(...)
Post-conditions:
- . . .
system operations
make New Sale()
enterItem (id, quantity)
Supplementary Specification
non-functional reqs, quality attributes
Operation Contracts
System Sequence Diagrams
requirements
Design Model
: Register
: ProductCatalog
: Sale
Design
enterItem (itemID, quantity)
spec = getProductSpec(itemID)
addLineItem(spec, quantity)

[La05]

Project Manager	**Robert Fornof**
Requirements/Analysis Team 1	**Hayden Chudy**
	He Zhang
	Weijian Jiang
	Jim Fletcher
	Xingyu Wang
Requirements/Analysis Team 2	**Chase Bryant**
	Hao Wu
	Rujie Yuan
	Kathleen Wilson
	Sumit Shrestha
Quality Assurance	**Christopher Camp**
	Calebe Hall
Configuration Management	**Matthew Cooper**
	Matthew Ngyuen
Testing/Security	**Adam Moore**
	Bradley Bittinger
	Leyue Wang
	Shanna Keith
	Adrianna Maniaci
Technology/Tool Integration	**Christopher Johnson**
	Glenn Bigelow



TEAM → assigned → ROLES and TASKS → produces → ARTIFACTS

Project Manager	Robert Fornof
Requirements/Analysis Team 1	Hayden Chudy
	He Zhang
	Weijian Jiang
	Jim Fletcher
	Xingyu Wang
Requirements/Analysis Team 2	Chase Bryant
	Hao Wu
	Rujie Yuan
	Kathleen Wilson
	Sumit Shrestha
Quality Assurance	Christopher Camp
	Calebe Hall
Configuration Management	Matthew Cooper
	Matthew Ngyuen
Testing/Security	Adam Moore
	Bradley Bittinger
	Leyue Wang
	Shanna Keith
	Adrianna Maniaci
Technology/Tool Integration	Christopher Johnson
	Glenn Bigelow

Defined for each Discipline

Discipline	Artifacts	Incep I1
Business Modeling	Domain Model	R
	Business rules	R
Requirements	Vision	R
	Use Case model	R
	Supplementary specification	R
	Stakeholder requests	R
	Glossary	R
	Prototypes and proof-of-concept	R
	Requirements management plan	R
	Software requirements specification	R
Configuration Management	Configuration management plan	R
Project Management	Iteration plan	R
	Software development plan (includes Phase plan)	R
	Business case	R
	Risk list	R
	Risk management plan	R
	Quality assurance plan	R
Environment	Development case	R

The management team is responsible for making these decisions

The class is responsible for accomplishing assigned activities and fulfilling roles

These relationships are clearly definable: they are part of a learning process for you and a teaching process for me

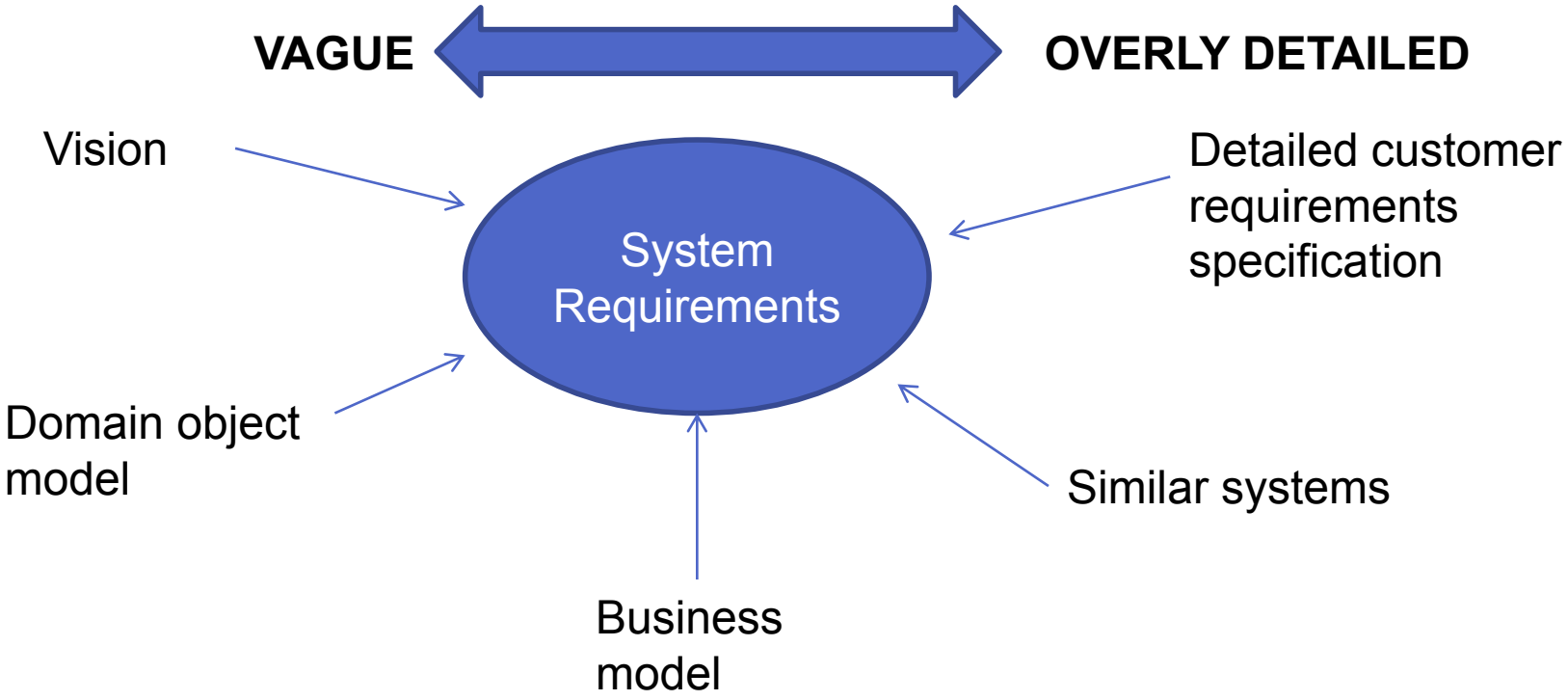
- Business Modeling
- Requirements
- Configuration Management
- Project Management
- Environment

- Develop a Domain model (part of requirements)
- Describe current business
- Identify business processes
  - Refine definitions, design realizations, refine role/responsibilities
- Explore process automation

- Objects or concepts: things in the system context that the system must manipulate or keep track of
- Events that transpire in the system context
- Capture as class models or (for small systems) as a glossary of terms
- Creates a common language for customer and developer
- Focus on domain modeling; defer system internal modeling to analysis, design, and implementation

- Business use case model
  - processes (use cases) and users (actors) in roles
  - represents system from a usage perspective and outlines how it provides value to its users
- Business object model
  - how each use case is realized by a set of workers who are using business entities and work units

- List candidate requirements
- Understand system context
- Capture functional requirements
- Capture non-functional requirements
- Validate requirements (not well-developed in RUP)



- Candidate features that could become requirements
  - Good ideas added to feature list
  - Features taken off list when they become formal requirements
- Planning values
  - Status
  - Cost
  - Priority
  - Risk

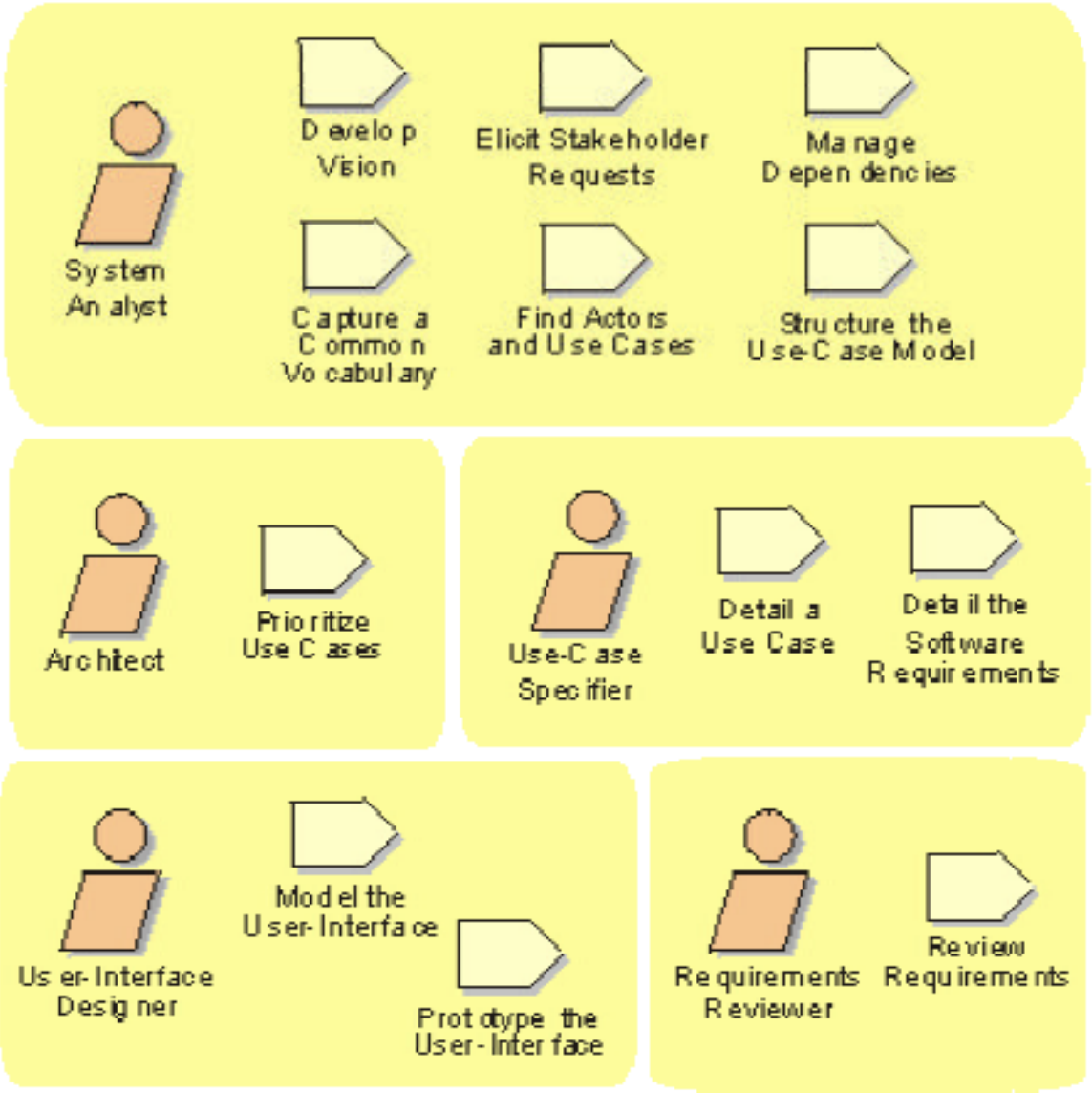
- Domain model
  - Identify and name important concepts and entities in the system context
  - Identify and name relations between domain objects
  - Glossary for now, possible classes in analysis and design workflows

- Business model
  - Domain object model PLUS
    - Processes/behaviors
    - Workers, their responsibilities and operations
  - Can be reflected in glossary, business model / workflow diagram, or in domain model itself

- Capture requirements as use cases
- Use case: a user's way of using the system
- When an actor (user or external subsystem) uses the system, the system performs a use case
- All use cases = all the things the system must do
- Capture user interfaces that support the use cases

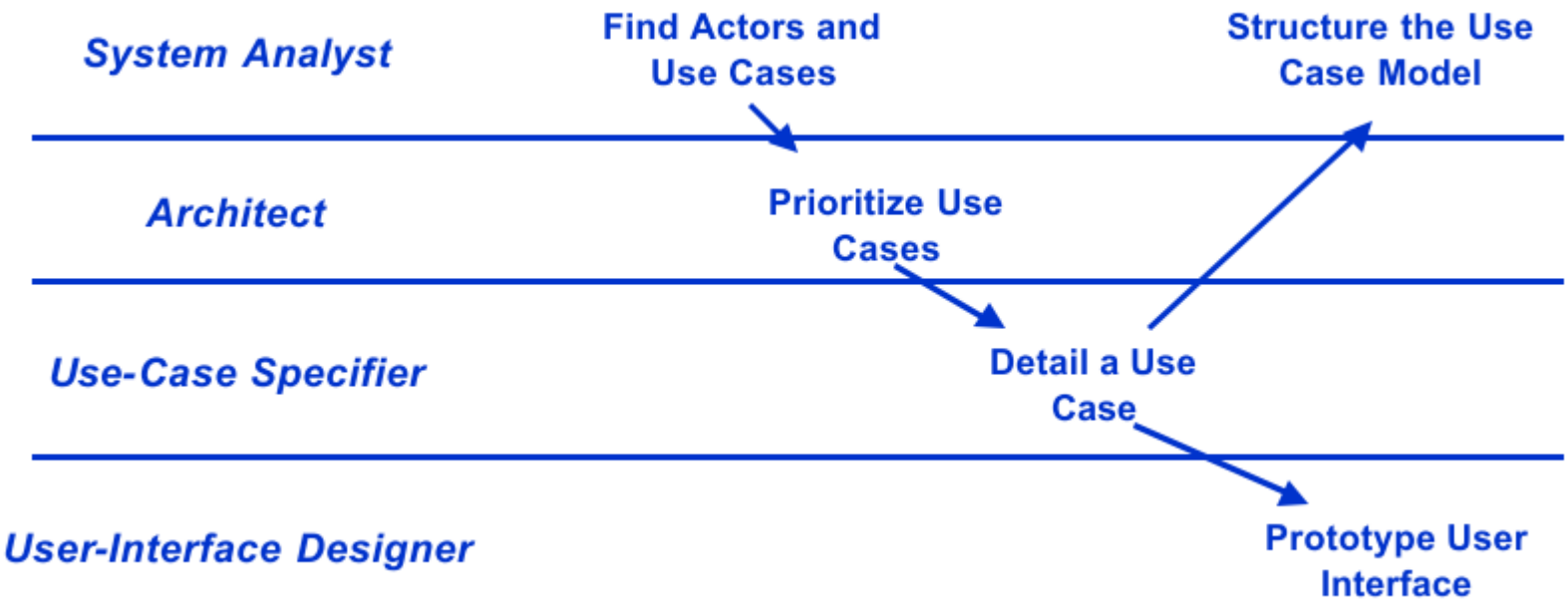
- System properties
  - Environmental or implementation constraints
    - e.g. must have remote access or must run on Linux or WinNT
  - Qualities (“-ilities”): performance, reliability, security, maintainability, extensibility, usability, etc.
- Tie to use cases or domain concepts, where possible
  - those that cannot be tied (they are general) are listed as supplementary requirements

- Roles that can be assigned to different tasks



- System analyst
  - Identify actors and use cases
  - Create complete and consistent set of use cases and requirements (but not for details of each individual use case)
  - Develop glossary to facilitate complete/consistent requirements set
- Use-case specifier
  - Detail one or more use cases
- User-interface designer
  - Define the “visual shape” of the user interface for one or more actors layout, behavior, inter-screen flow
- Architect
  - Describe architectural view of use-case model

Core Requirements Workflow



- Manage change requests
- Plan project configuration and control
- Create project CM environments
- Monitor and report configuration status
- Change and deliver configuration items
- Manage baseline and releases

- Evaluate project scope and risk
- Develop software development plan
- Monitor and control project
- Plan for next iteration
- Manage iteration

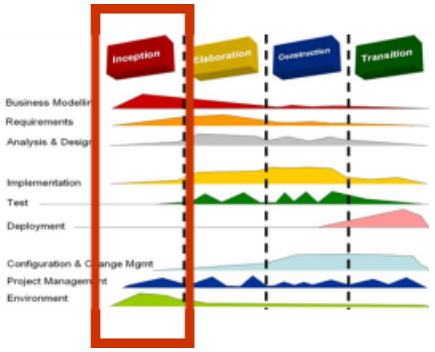
- Roles:
  - Process Engineer
    - Development case, project-specific guidelines, artifact tailoring
  - Tool specialist
    - Development tools and practices
  - Systems administration
    - Development infrastructure
  - Technical writers
    - Manual styleguide

- Tasks
  - Prepare environment for project
  - Prepare environment for iteration

REFERENCE MATERIAL

Some Artifacts that Start in Inception

- **Vision and business case**
  - Describes high-level goals and constraints.
- **Use Case model**
  - Describes functional requirements and related non-functional requirements.
- **Supplementary specification**
  - Describes other requirements
- **Stakeholder requests**
  - Records interviews/requests from the stakeholder
- **Glossary**
  - Key domain terminology



- **Risk list and Risk Management Plan**
  - Describes business, technical, resource and schedule risks and ideas for their mitigation or response.
- **Prototypes and proof-of-concepts**
- **Iteration plan**
  - Describes what to do in the first elaboration iteration
- **Phase Plan & Software development Plan**
  - Guess for elaboration phase duration. Tools, people, education and other resources.
- **Development Case**
  - Description of the customized UP steps and artifacts for this project

[La05]

- An executive overview of the envisioned system
  - Summarizing other requirements documents
- Content
  - Positioning (business opportunity, problem statement)
  - Stakeholder descriptions
    - Both users and non-users, their goals and problems
  - Product overview (summary of benefits, assumptions, cost)
  - Summary of system features (about 10 things the system *does*)
    - More abstractly than a list of use cases
  - Other requirements and constraints
- Format: see RUP templates
  - rup_vision.dot
  - rup_buscs.dot

- **Vision and business case**
  - Describes high-level goals and constraints.
- **Use Case model**
  - Describes functional requirements and related non-functional requirements.
- **Supplementary specification**
  - Describes other requirements
- **Stakeholder requests**
  - Records interviews/requests from the stakeholder
- **Glossary**
  - Key domain terminology
- **Risk list and Risk Management Plan**
  - Describes business, technical, resource and schedule risks and ideas for their mitigation or response.
- **Prototypes and proof-of-concepts**
- **Iteration plan**
  - Describes what to do in the first elaboration iteration
- **Phase Plan & Software development Plan**
  - Guess for elaboration phase duration. Tools, people, education and other resources.
- **Development Case**
  - Description of the customized UP steps and artifacts for this project

[La05]

- Content
  - Requirements that don't fit into the use case format
  - From all FURPS+ categories (more to come on this)
  - Some people include UI mockups or storyboards
    - These are *not* requirements and should be marked "notional" or omitted
  - A list of "shall statements" about the system and its features
  - Arranged topically
  - Consider using quality attribute scenarios
- Format: see RUP templates
  - rup_sspec.dot

- **Vision and business case**
  - Describes high-level goals and constraints.
- **Use Case model**
  - Describes functional requirements and related non-functional requirements.
- **Supplementary specification**
  - Describes other requirements
- **Stakeholder requests**
  - Records interviews/requests from the stakeholder
- **Glossary**
  - Key domain terminology
- **Risk list and Risk Management Plan**
  - Describes business, technical, resource and schedule risks and ideas for their mitigation or response.
- **Prototypes and proof-of-concepts**
- **Iteration plan**
  - Describes what to do in the first elaboration iteration
- **Phase Plan & Software development Plan**
  - Guess for elaboration phase duration. Tools, people, education and other resources.
- **Development Case**
  - Description of the customized UP steps and artifacts for this project

[La05]

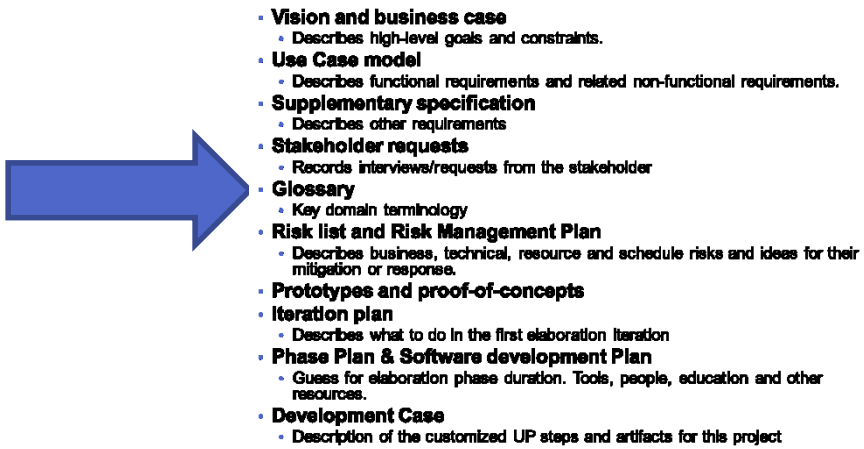
- Helps formulate “interviews” with the stakeholder to begin the requirements elicitation process
- Format: see RUP templates
  - rup_stkreq.dot

- **Vision and business case**
  - Describes high-level goals and constraints.
- **Use Case model**
  - Describes functional requirements and related non-functional requirements.
- **Supplementary specification**
  - Describes other requirements
- **Stakeholder requests**
  - Records interviews/requests from the stakeholder
- **Glossary**
  - Key domain terminology
- **Risk list and Risk Management Plan**
  - Describes business, technical, resource and schedule risks and ideas for their mitigation or response.
- **Prototypes and proof-of-concepts**
- **Iteration plan**
  - Describes what to do in the first elaboration iteration
- **Phase Plan & Software development Plan**
  - Guess for elaboration phase duration. Tools, people, education and other resources.
- **Development Case**
  - Description of the customized UP steps and artifacts for this project

[La05]

- A list of noteworthy terms and their definitions
- The UP glossary also plays the role of data dictionary, recording *meta-data* used by the system
- Format: see RUP templates
  - rup_gloss.dot

Term	Definition	Format	Validation rules	Aliases
ISBN	Numeric code that identifies a book	10-digit code with subparts	Digit 10 is a check digit	International Standard Book Number



[La05]

- Document the risk management planning process
- Should include considerations for security, technology, resources, schedule
- Mitigation strategy: how will risk be monitored, tracked, addressed?
- Format: see RUP templates
  - rup_rmpln.dot, rup_rsklst.dot, rup_rskpln.dot

- **Vision and business case**
  - Describes high-level goals and constraints.
- **Use Case model**
  - Describes functional requirements and related non-functional requirements.
- **Supplementary specification**
  - Describes other requirements
- **Stakeholder requests**
  - Records interviews/requests from the stakeholder
- **Glossary**
  - Key domain terminology
- **Risk list and Risk Management Plan**
  - Describes business, technical, resource and schedule risks and ideas for their mitigation or response.
- **Prototypes and proof-of-concepts**
- **Iteration plan**
  - Describes what to do in the first elaboration iteration
- **Phase Plan & Software development Plan**
  - Guess for elaboration phase duration. Tools, people, education and other resources.
- **Development Case**
  - Description of the customized UP steps and artifacts for this project

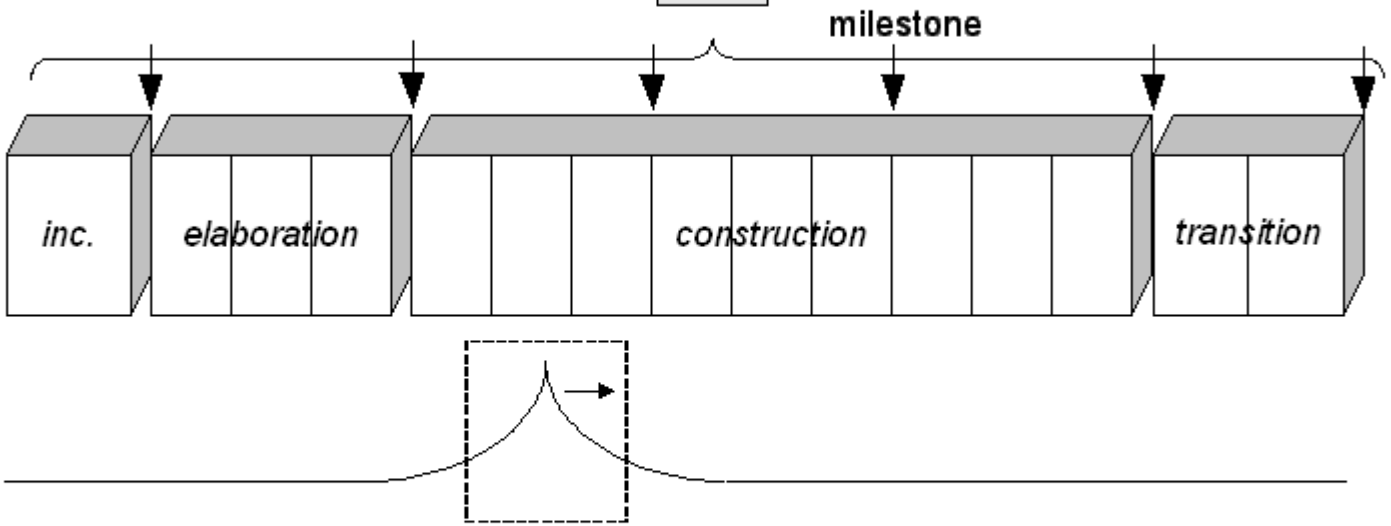
[La05]

Phase Plan versus Iteration Plan

Short; a few pages. Estimates phase and milestone end dates, and their objectives.

Phase Plan

Format: see RUP templates
rup_itpln.dot
rup_sdpln.dot



Detailed planning in an Iteration Plan is like a rolling wave that is only highly specific around the present and the near future (for example, the next iteration).

Iteration Plan

- **Vision and business case**
  - Describes high-level goals and constraints.
- **Use Case model**
  - Describes functional requirements and related non-functional requirements.
- **Supplementary specification**
  - Describes other requirements
- **Stakeholder requests**
  - Records interviews/requests from the stakeholder
- **Glossary**
  - Key domain terminology
- **Risk list and Risk Management Plan**
  - Describes business, technical, resource and schedule risks and ideas for their mitigation or response.
- **Prototypes and proof-of-concepts**
- **Iteration plan**
  - Describes what to do in the first elaboration iteration
- **Phase Plan & Software development Plan**
  - Guess for elaboration phase duration. Tools, people, education and other resources.
- **Development Case**
  - Description of the customized UP steps and artifacts for this project

[La05]

- Documents *how* RUP will be tailored for this project
- Details artifacts, review processes, tools, templates, references to other plans
- Format: see RUP templates
  - rup_devcs.dot

[La05]

- **Vision and business case**
  - Describes high-level goals and constraints.
- **Use Case model**
  - Describes functional requirements and related non-functional requirements.
- **Supplementary specification**
  - Describes other requirements
- **Stakeholder requests**
  - Records interviews/requests from the stakeholder
- **Glossary**
  - Key domain terminology
- **Risk list and Risk Management Plan**
  - Describes business, technical, resource and schedule risks and ideas for their mitigation or response.
- **Prototypes and proof-of-concepts**
- **Iteration plan**
  - Describes what to do in the first elaboration iteration
- **Phase Plan & Software development Plan**
  - Guess for elaboration phase duration. Tools, people, education and other resources.
- **Development Case**
  - Description of the customized UP steps and artifacts for this project