



# Requirements Engineering: Wrap-up CECS 590

# Recap time!

- Requirements Management
  - What tasks does it entail?
  - Change Management – how?
  - Risk Management – how?



# Requirements Engineering – Outline

- WHY do we need Requirements Engineering and what is it?
- Principles: Definitions, process, roles, problem/solution view, artifact orientation
- System Models: Decomposition and abstraction, system views
- Frameworks: What reference structures can I use for requirements?
- Business Case Analysis: Why are we building this system?
- Stakeholders: Who are the people to talk to about requirements?
- Goals and Constraints: What are the major objectives for the system?
- System Vision: What exactly do we want to achieve?
- Domain Models: What are the surrounding systems ours interacts with?
- Usage Models: How will the system interact with the user?
- Software quality models: How to determine the quality characteristics?
- Quality requirements: How to specify which qualities need to be met?
- Process requirements: How to specify constraints for development?
- Towards a system specification: How to hand over to design?
- Quality assurance: How to ensure that RE is done in a good way?
- Change management: How to evolve requirements?
- Wrap-up: How to put it all together



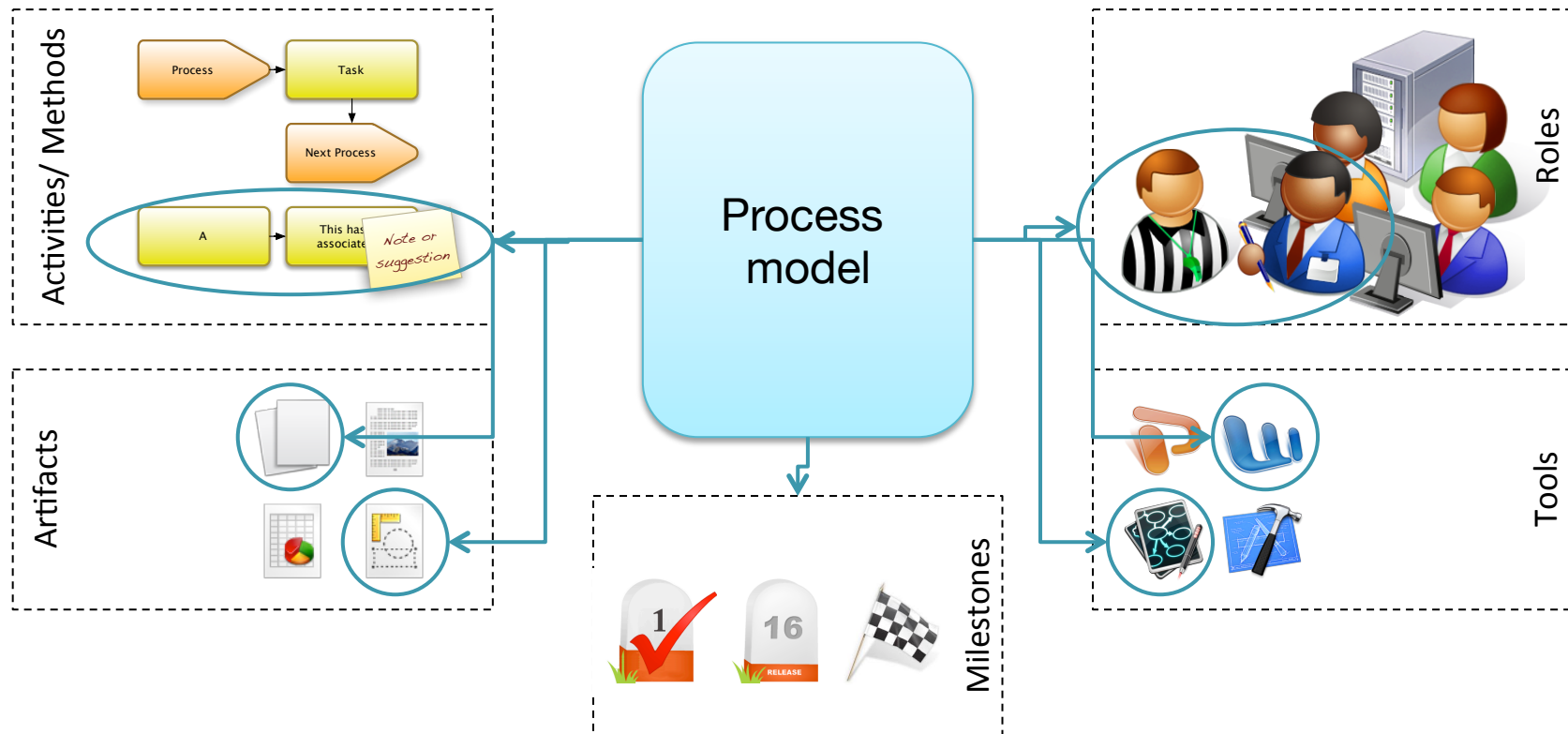
# Project plan: What to define?

Artwork: Jonathan Harris

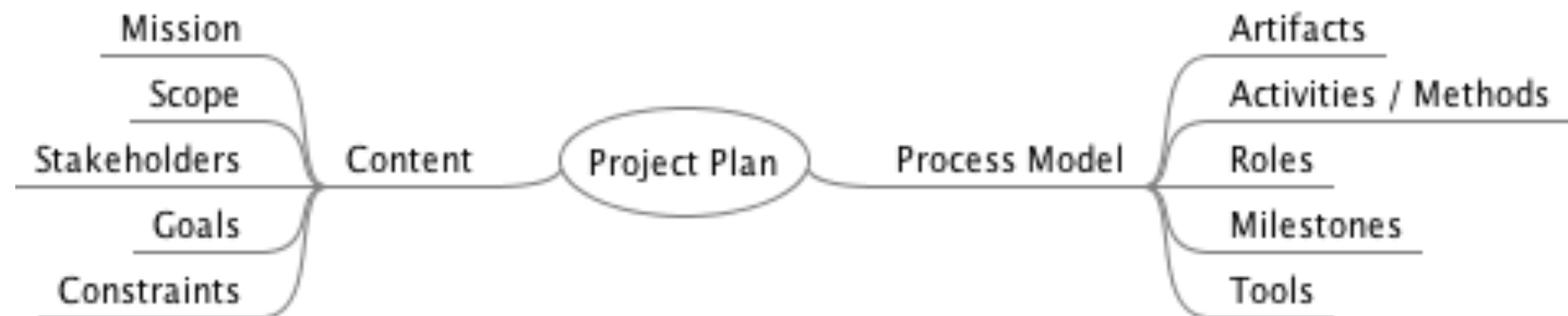


# Project plan: What to define?

## Infrastructure plus project content.

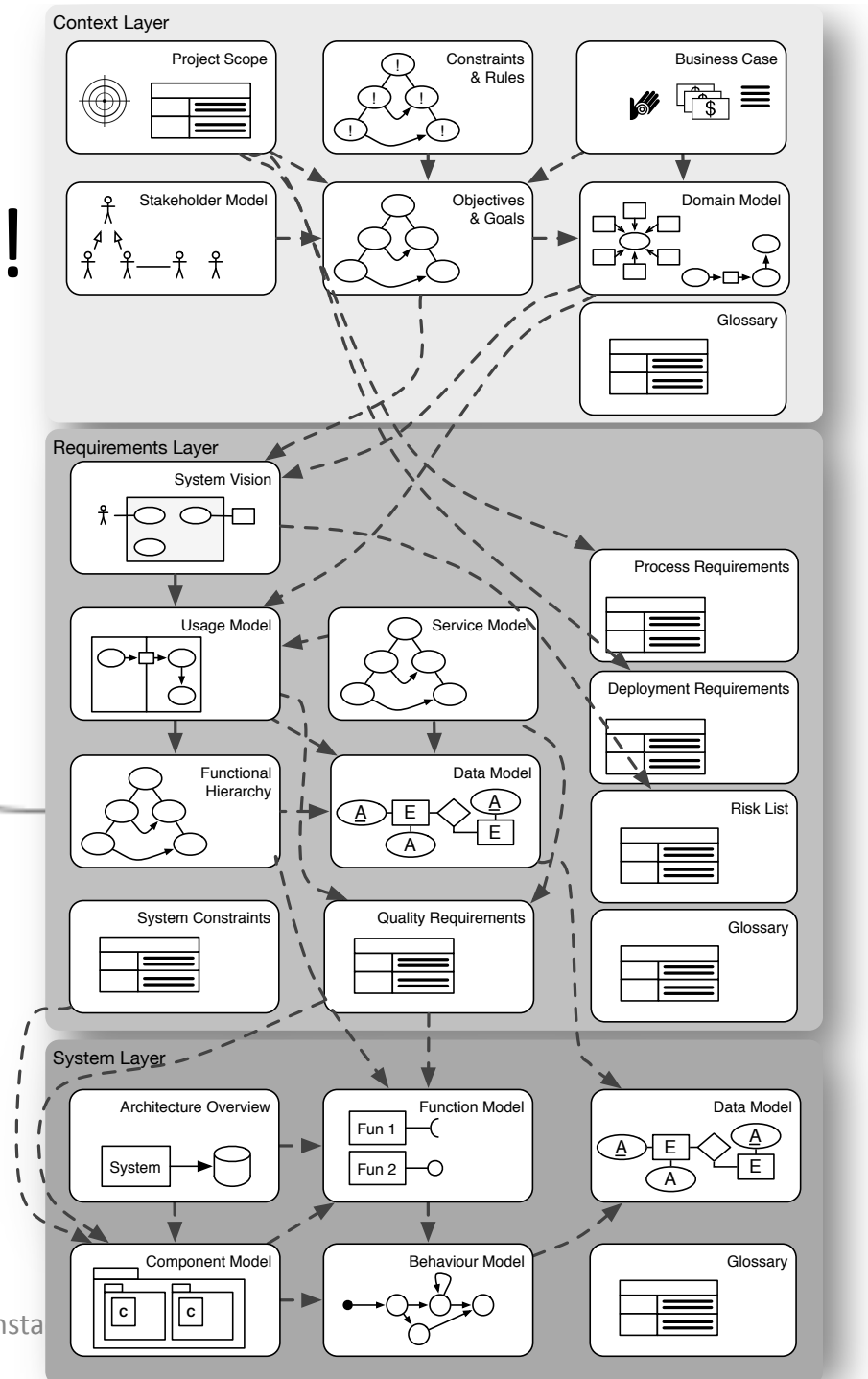
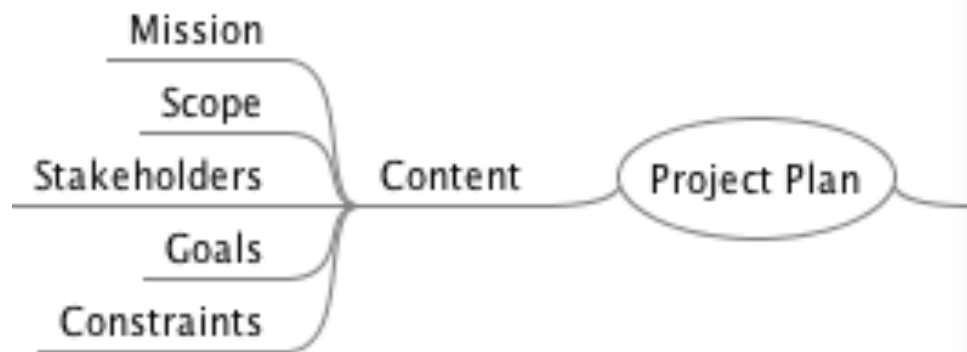


# Fill in...



# Sounds familiar?

## Guess what... artifacts!



# Let's do it: Infrastructure

1. Artifacts
2. Activities
3. Roles
4. Milestones
5. Tools



# Let's do it: Project Contents

1. Mission
2. Scope
3. Stakeholders
4. Goals
5. Constraints

# Let's do it: Budget

It's spreadsheet o'clock!!

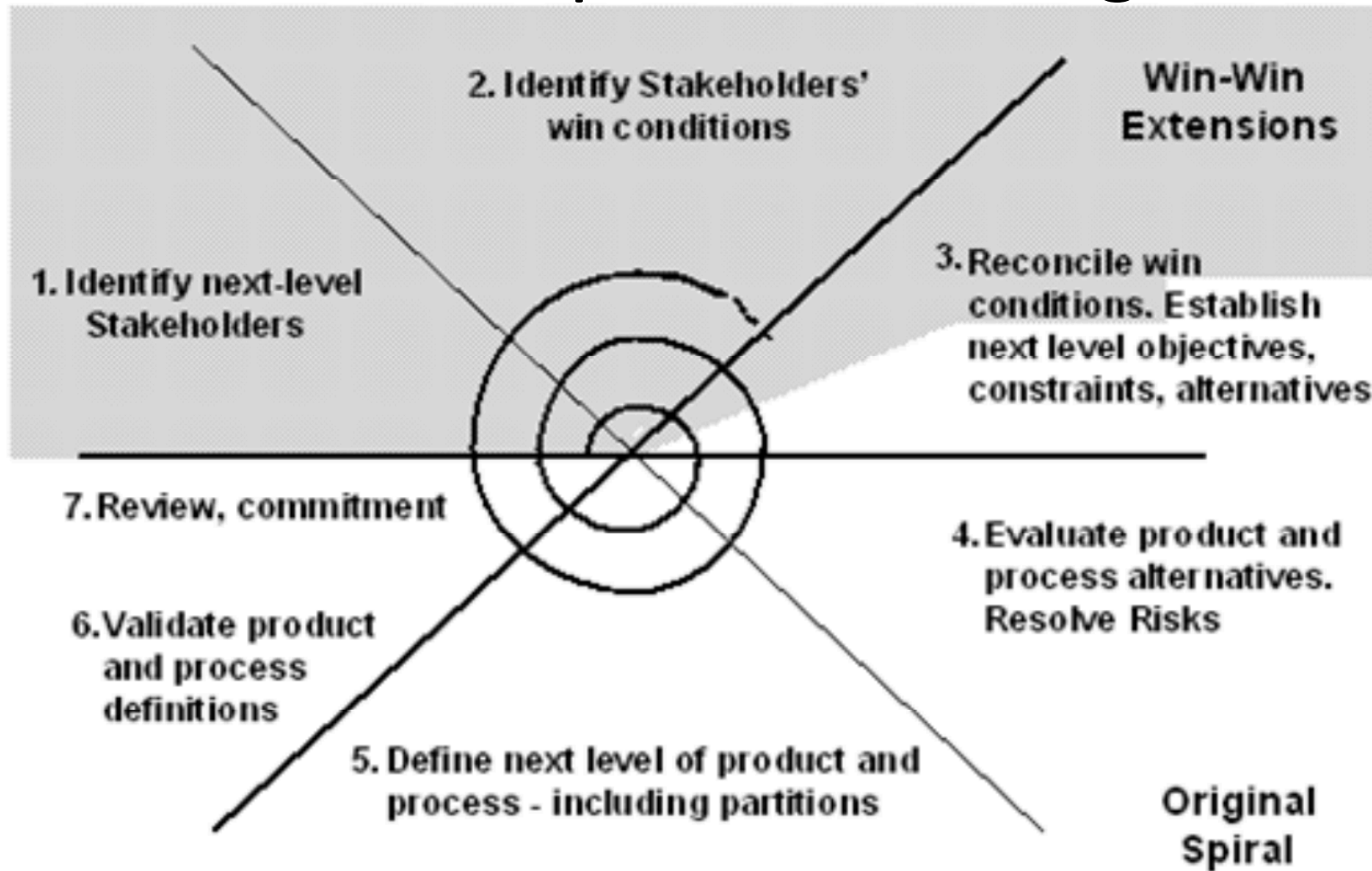
1. (Estimation of required hardware)
2. (Estimation of required software)
3. Estimation of development work



# Then: Define for Project Management

- Change Management
- Risk Management
- Validation & verification
- Version & configuration management
- Claim management
- Support for distributed RE

# Details: Requirements Negotiation



**Figure 8.1: The WinWin Spiral Model of Software Engineering includes front-end activities (gray) that show where objectives, constraints, and alternatives come from. This lets users more clearly identify the rationale involved in negotiating win conditions for the product.**

