CS 5704: Software Engineering

Connecting the Software and Usability Engineering Life Cycles

Dr. Pardha S. Pyla

Interactive Software Systems

Interactive Software Systems

User interface

Interactive Software Systems

User interface

Functional core

Interactive Software Systems

User interface

psychology

cognition

design guidelines

task analysis

Functional core

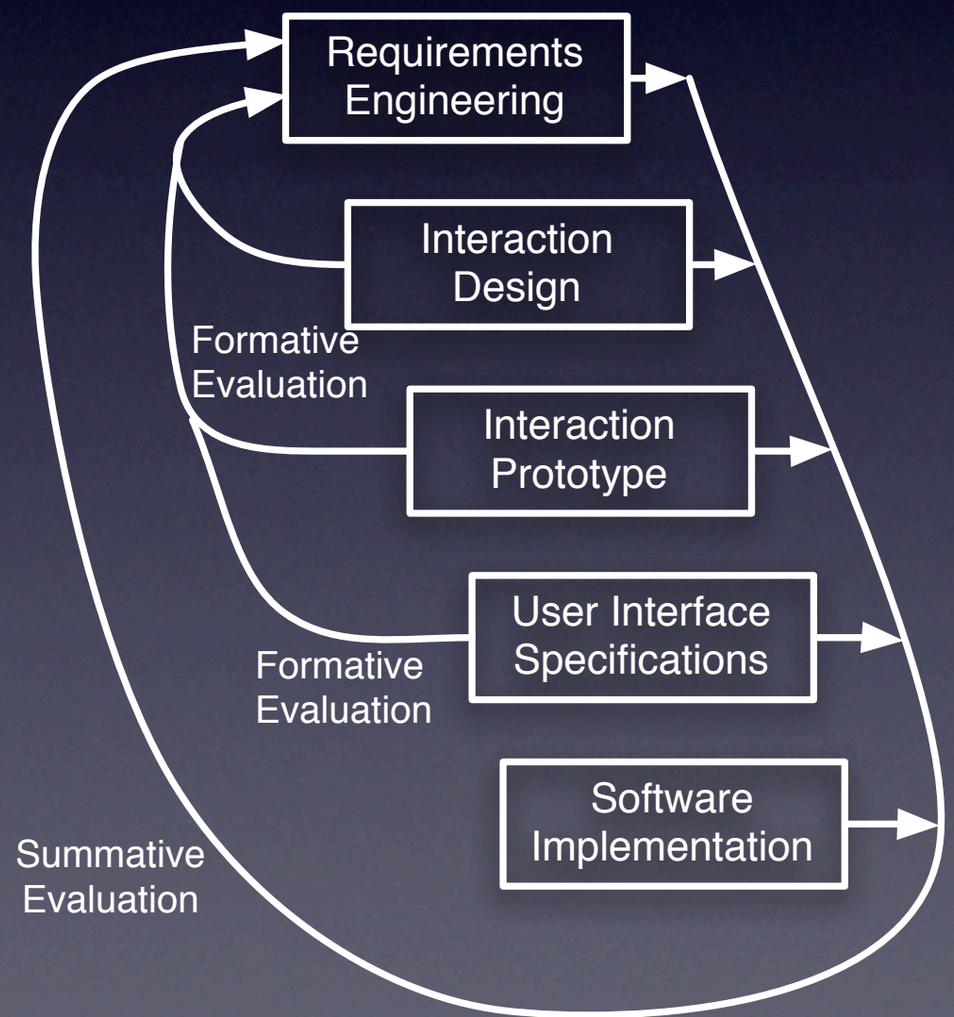
Interactive Software Systems

User interface	Functional core
psychology	algorithms
cognition	data structures
design guidelines	software architectures
task analysis	database management

Development Processes

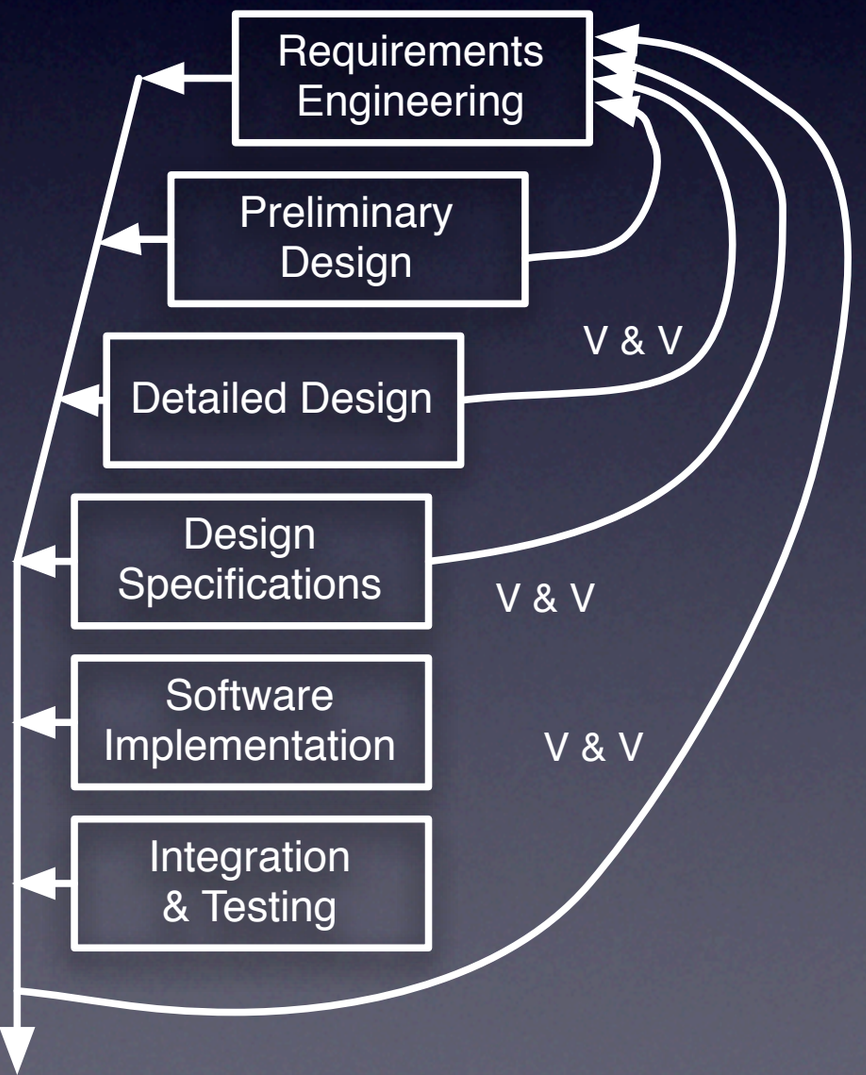
Usability engineering

(user interface)

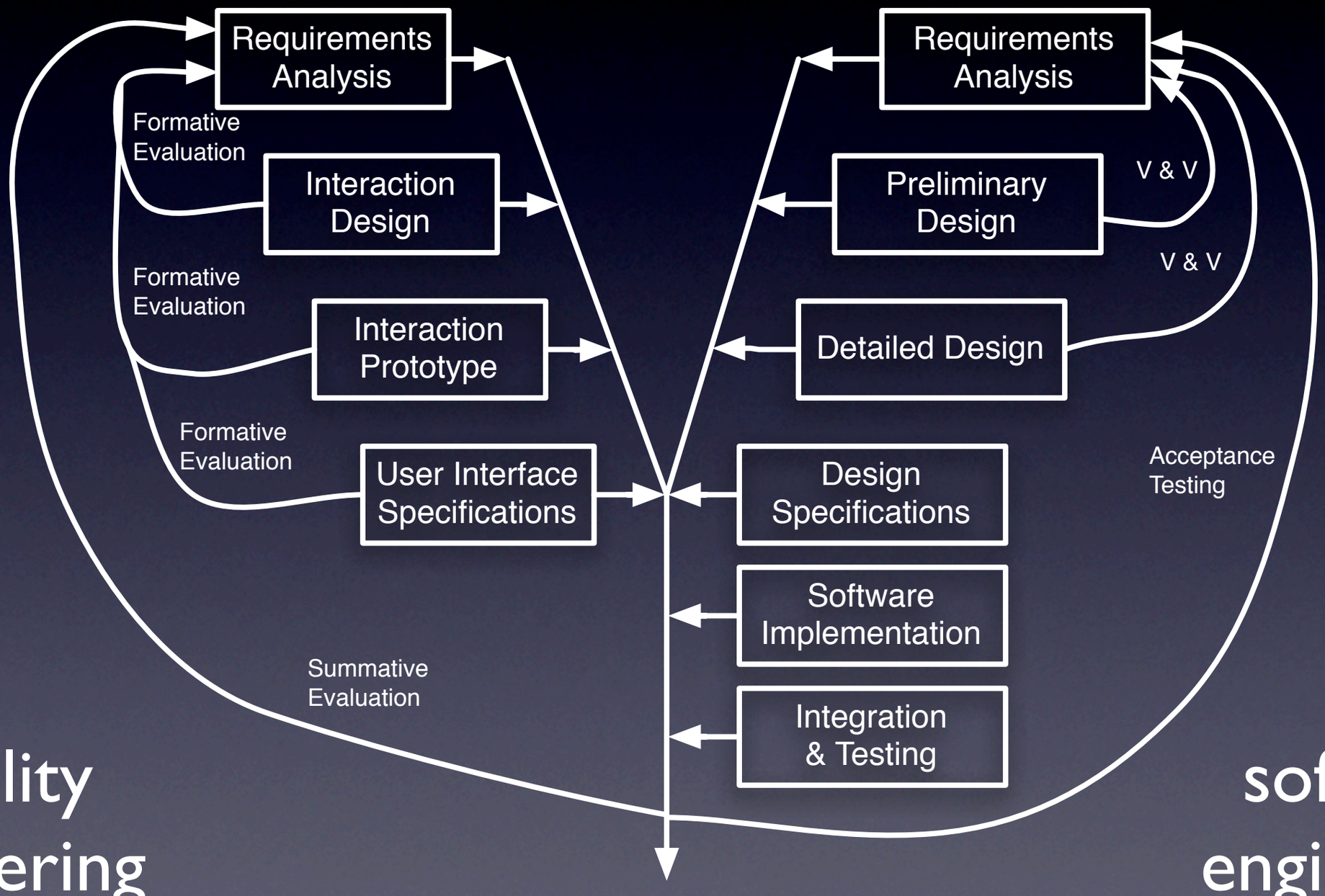


Software engineering

(functional core)



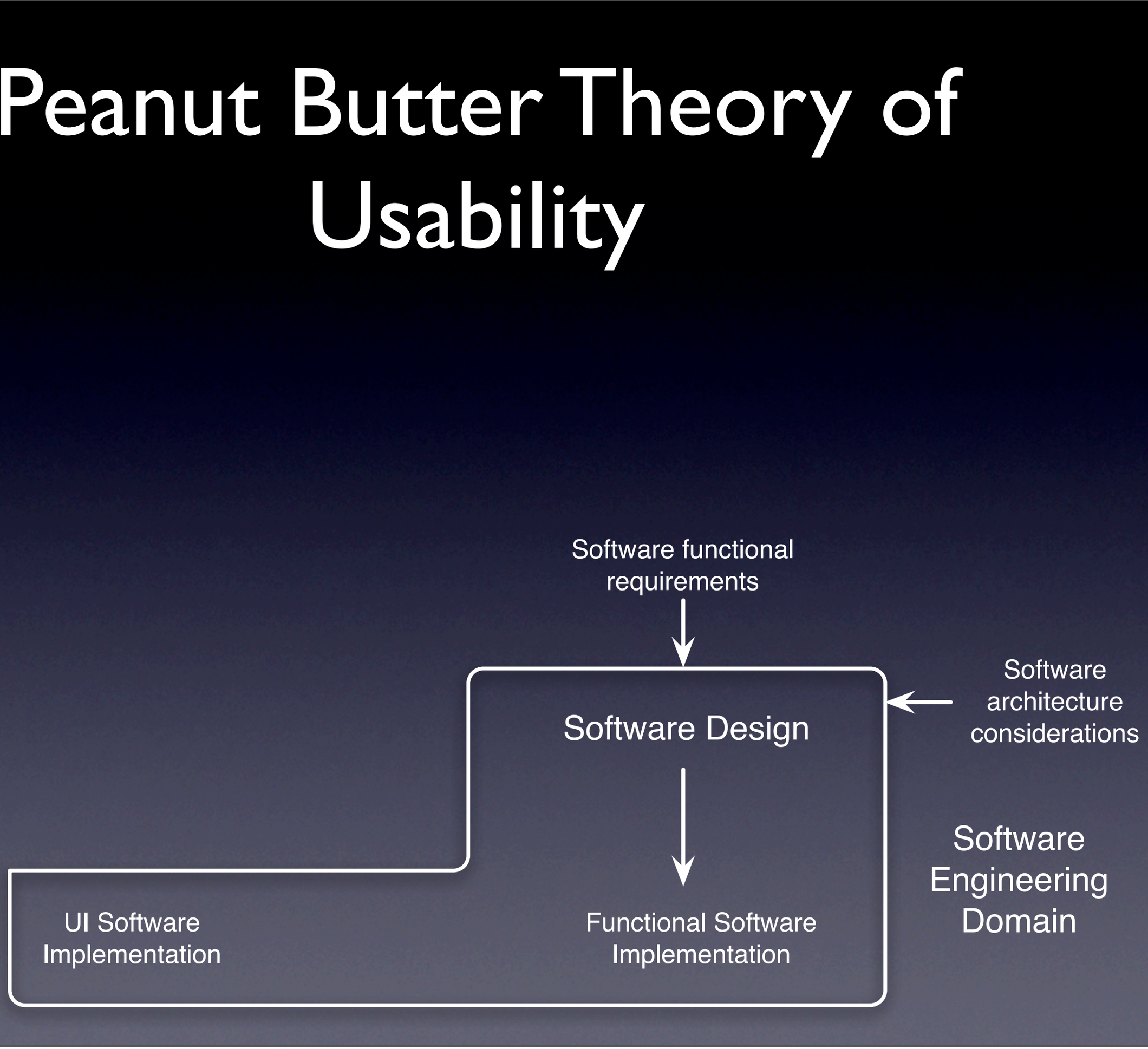
Missing Connections



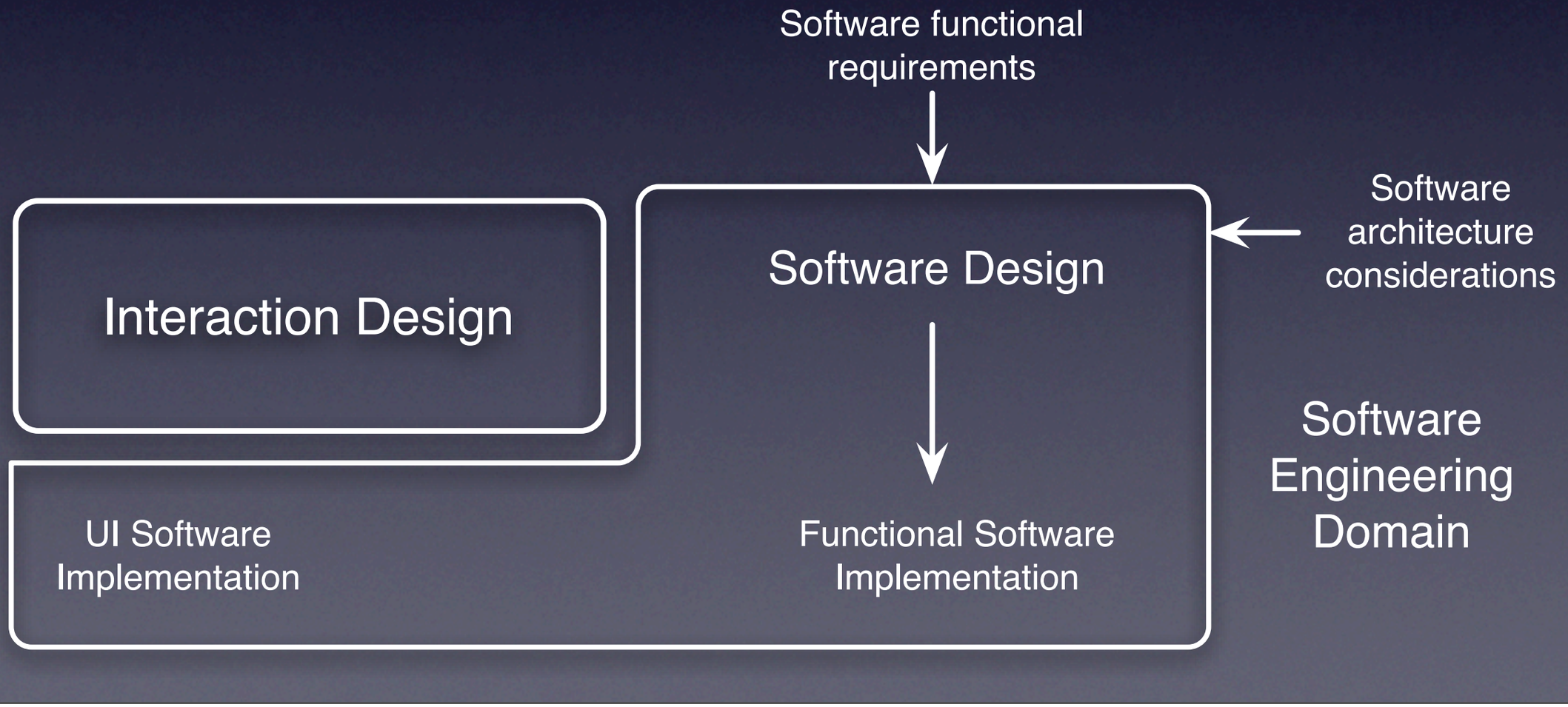
usability engineering

software engineering

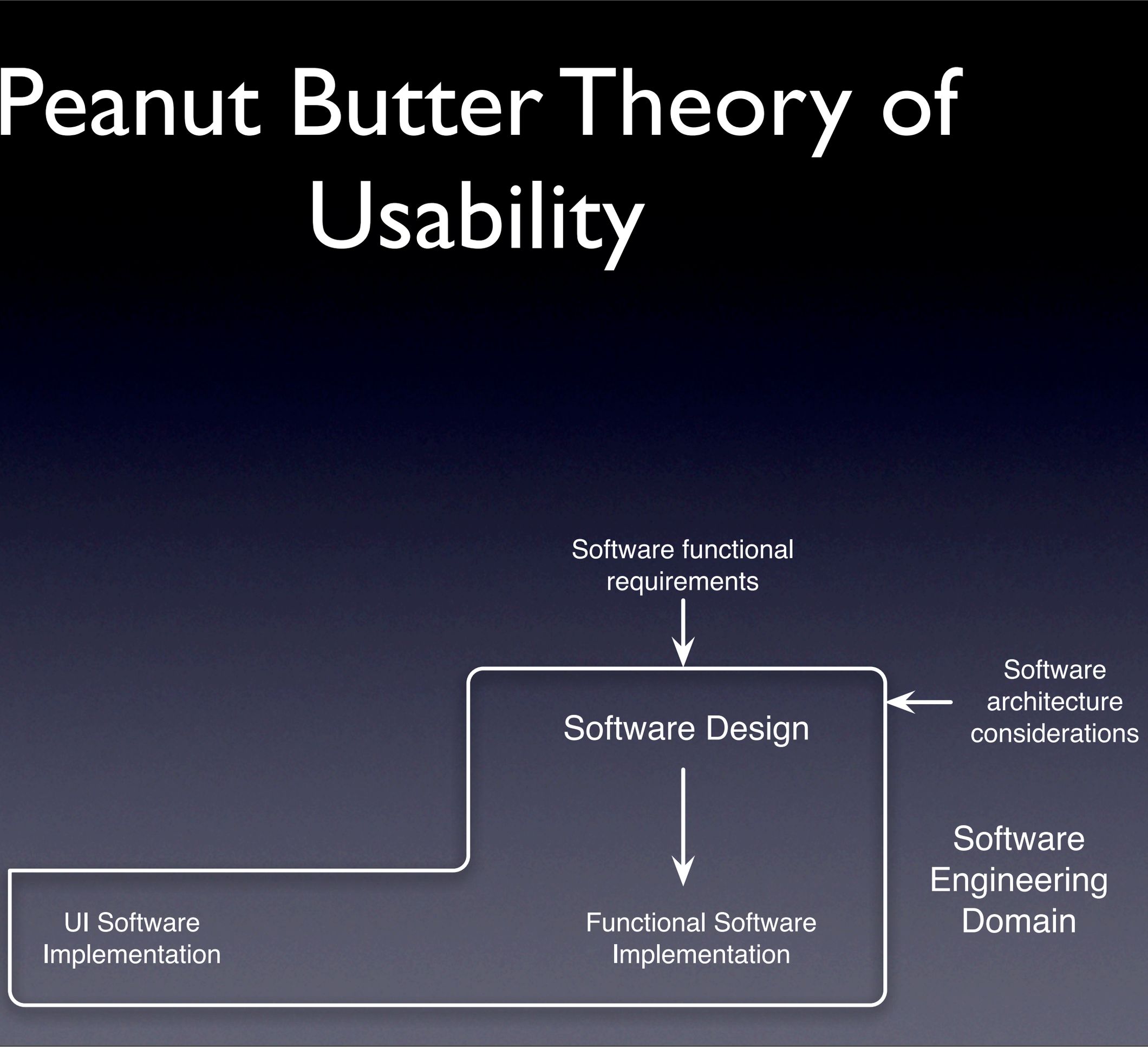
Peanut Butter Theory of Usability

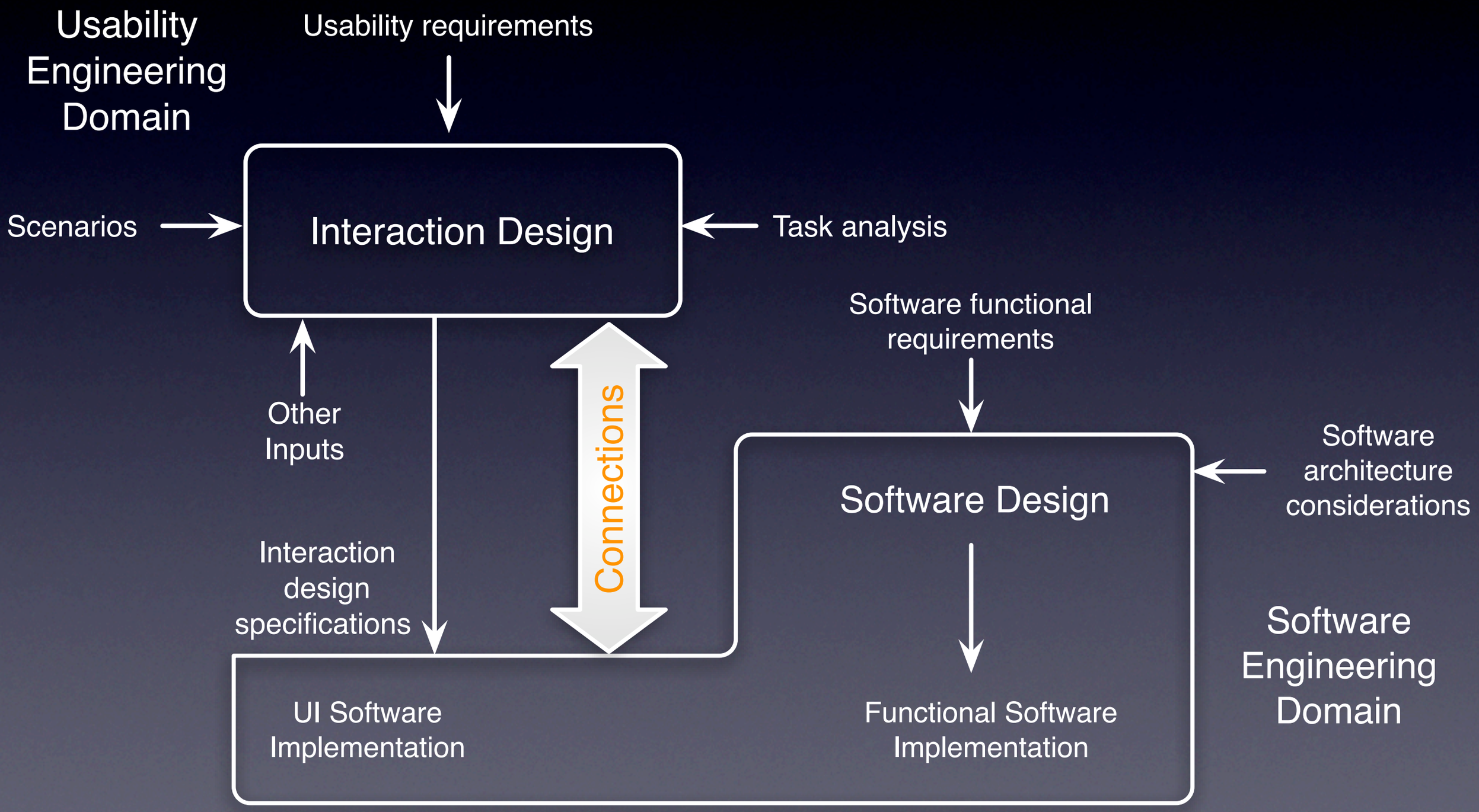


Peanut Butter Theory of Usability



Peanut Butter Theory of Usability




Usability Engineering Domain
Usability requirements
Scenarios
Interaction Design
Task analysis
Other Inputs
Connections
Software functional requirements
Software architecture considerations
Software Design
Interaction design specifications
UI Software Implementation
Functional Software Implementation
Software Engineering Domain

Why Should We Care?

- Chaos report by the Standish Group
  - US spent $250 billion on software applications (N=350)
  - 31% projects cancelled before completion
  - 53% costed 189% of their original estimates
  - 16% were on-time and on-budget (larger companies only 9%)
- Project challenged factors
  - Lack of user involvement
  - Incomplete requirements

Overarching Problem

- There is a need for
  - SE and UE life cycles to be *connected* through development environments to support communication, and
  - appropriate curricula to teach about these connections

Lack of Development Environments for Connections

- Development efforts rely on structure of development environments
  - Environments embody particular methodologies
- Current software development environments do not address or support usability engineering
  - Reducing the ability to react to change
  - Reduced quality of the overall system

Lack of Appropriate Curricula

- A consequence of decades of reductionist approach
  - Reducing complex problems into isolated modules
  - Producing "vertical" experts, suitable for assembly-line jobs
- Missing out on
  - Integrationist approaches that address relations among modules
  - Producing "holistic" experts, suitable for today's complex problems
- "*Complex problems require multidimensional and interdisciplinary analysis and solutions*"

Research Approach: Development Environments

- Ripple Description Model
  - formally and abstractly charted development space
- Ripple Implementation Framework
  - described how model translates into a real tool
- Ripple Implementation Instance
  - to suit the requirements/constraints of classroom setting
- Investigated communication factors via Ripple instance

Research Approach: Pedagogy

- Created a cross-pollinated SE-UE course
  - SE and UE classes together with joint teams
    - simulated different conditions of interactive-software development
    - demonstrated dynamics of SE-UE interaction via end-of-semester symposium

Ripple Description Model

Form

wa_i =

<wa-symbolic-name, LC-type, wa-type, wa-technique, wp-affected, developer-role, developer-name>

Example

wa_1 =

<CW-1, UE, formative usability evaluation, cognitive walkthrough, usability problem list, usability evaluator, John Doe>

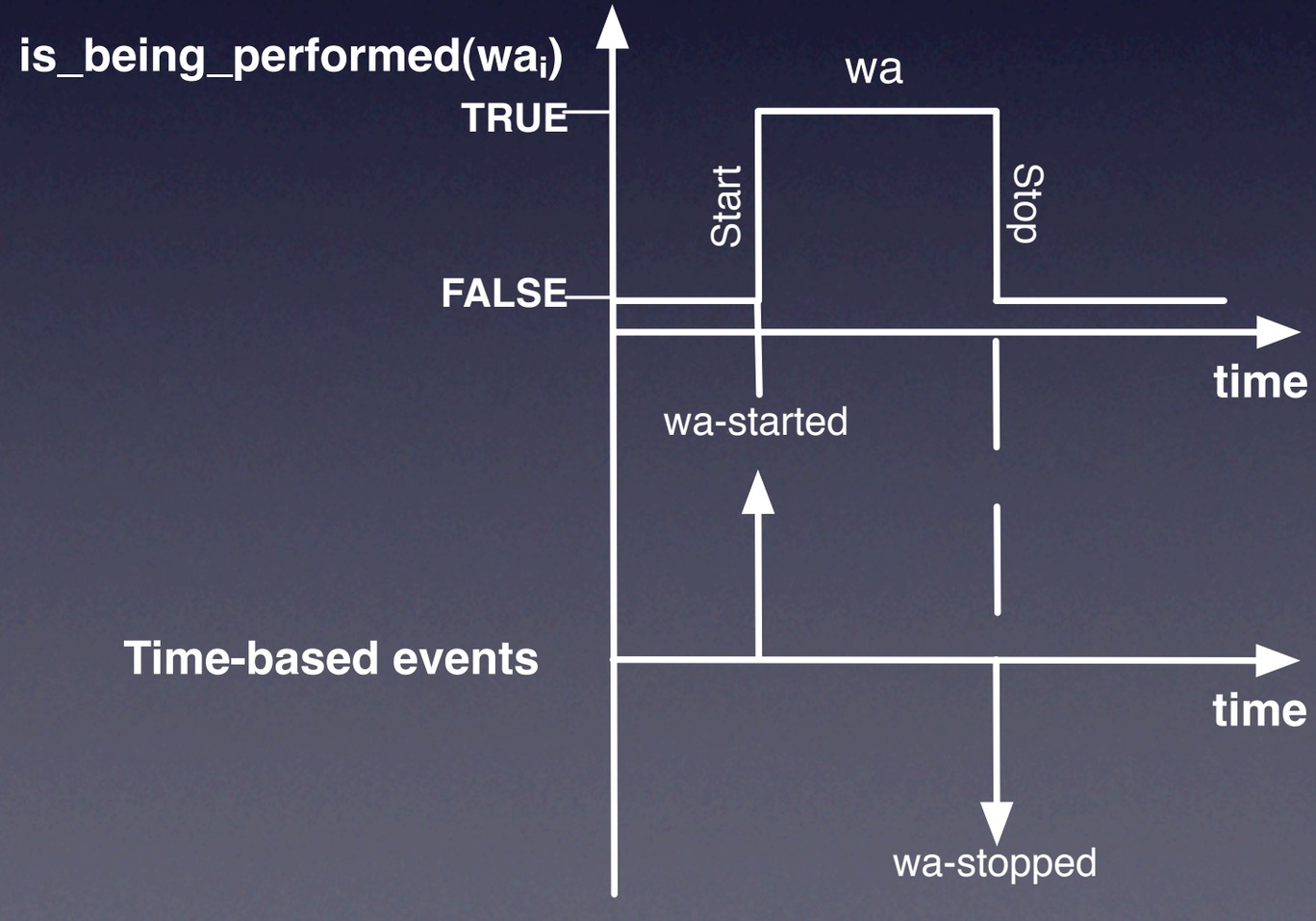
Functions

Selection:

$Y = \{wa \mid \prod_{wp\text{-}affected}(wa_1) = usability\ problem\ list\}$

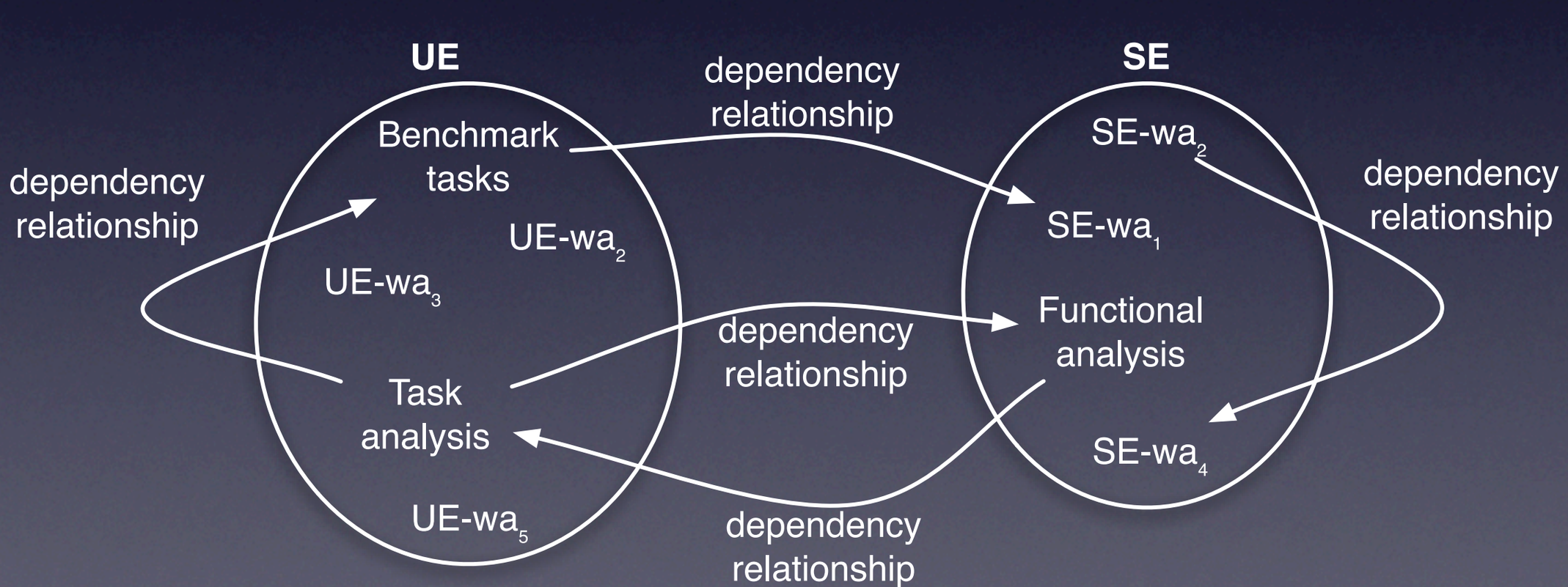
State Variables To Describe Events

$$is\_being\_performed(wa_i) = \begin{cases} true, \text{ if and only if } wa_i \text{ is currently being performed} \\ false, \text{ otherwise} \end{cases}$$



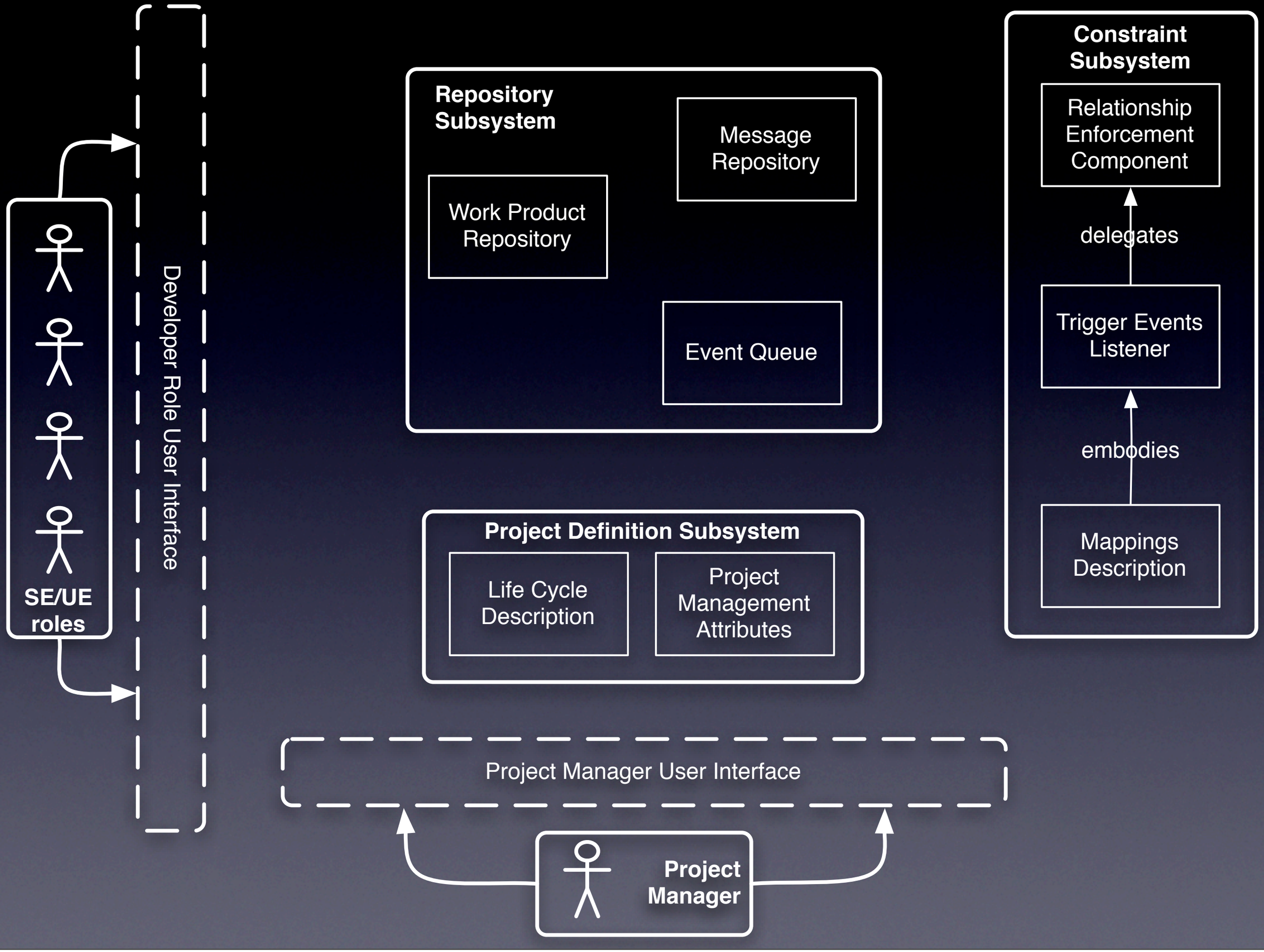
Dependencies and Mappings

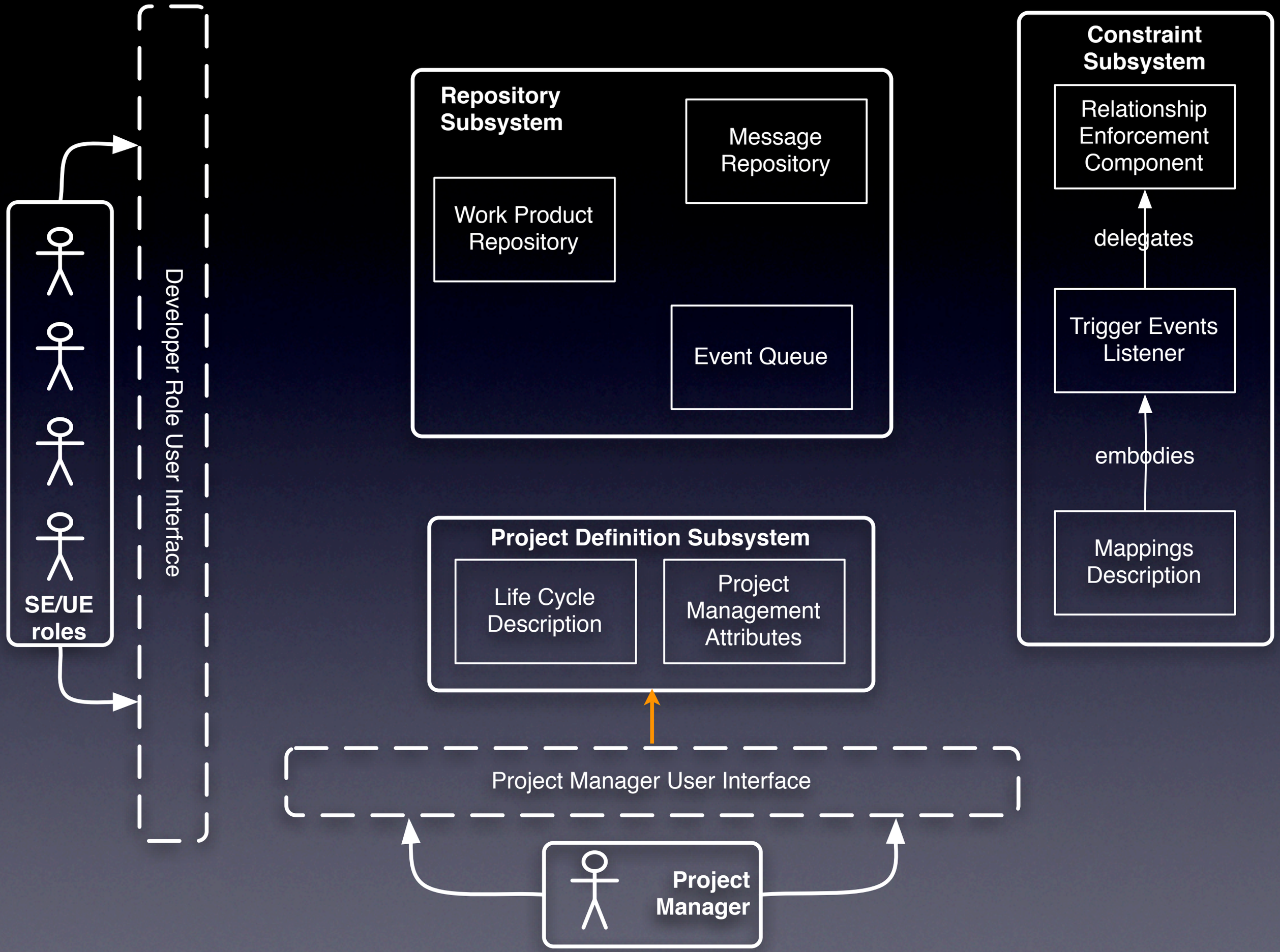
m = <wa_{source}, *te*, wa_{target}, *r*>

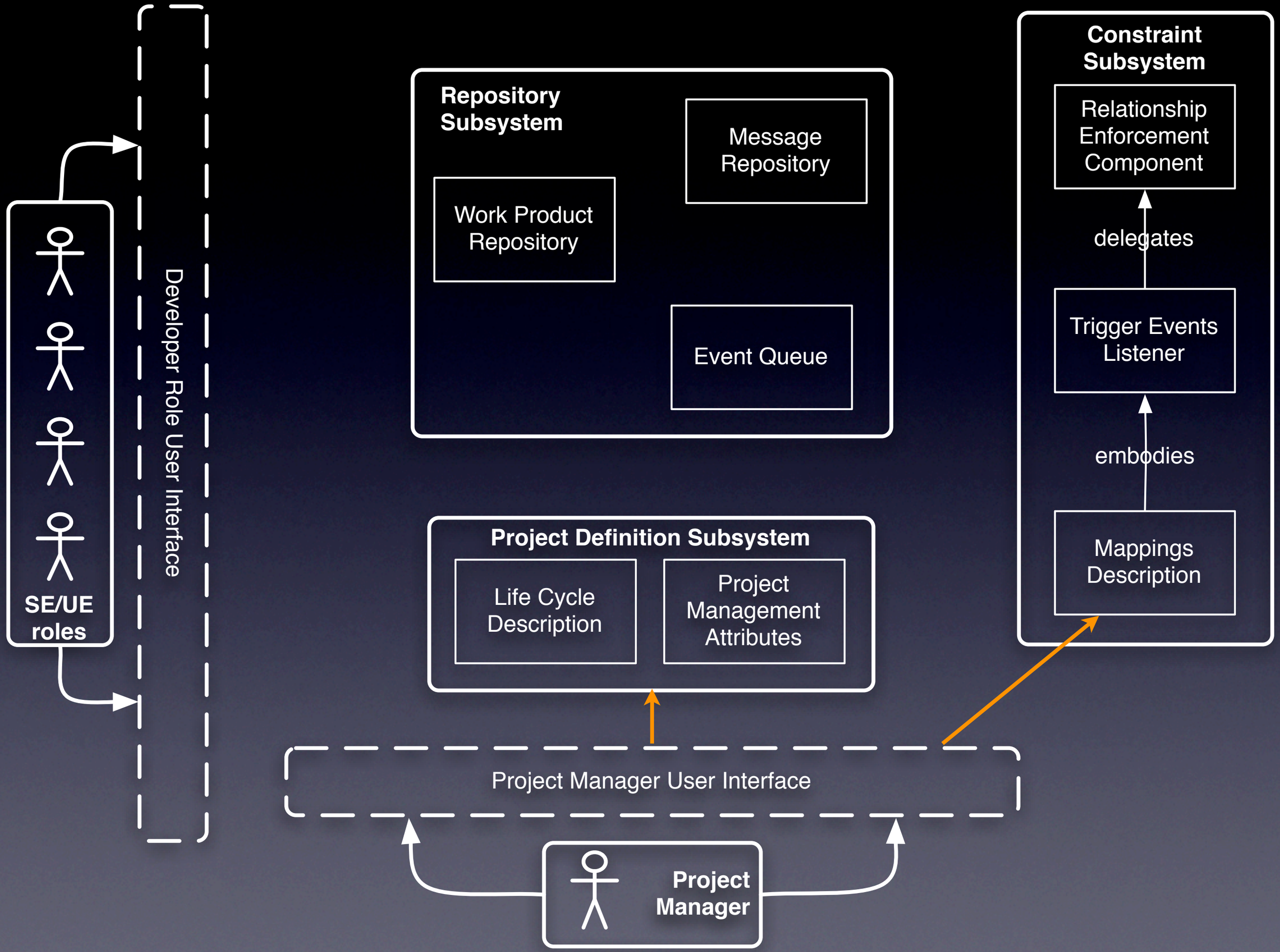


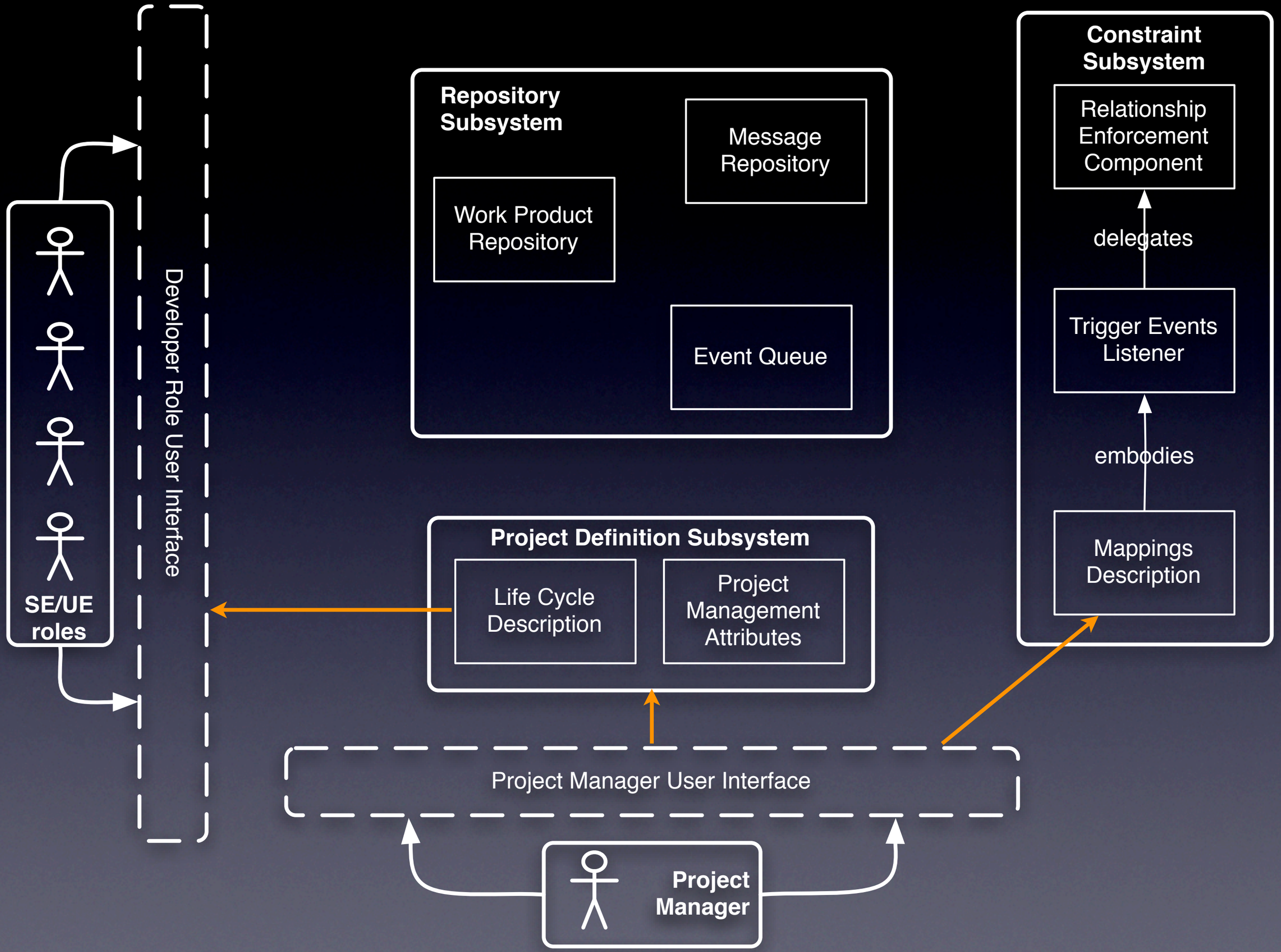
Ripple Implementation Framework

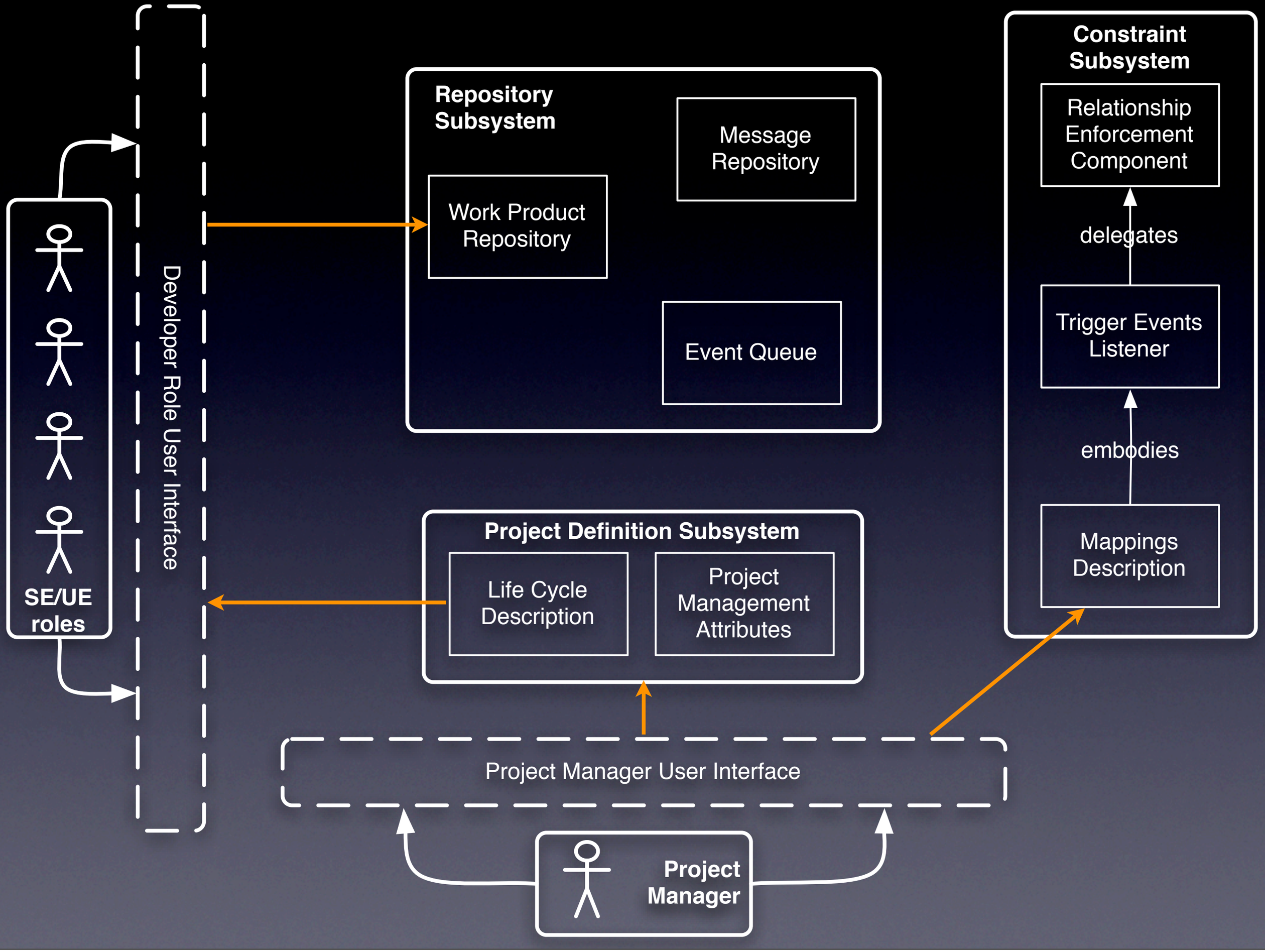
- Reification of the Description Model
- Describes the environment and tool support involved in interactive-systems development

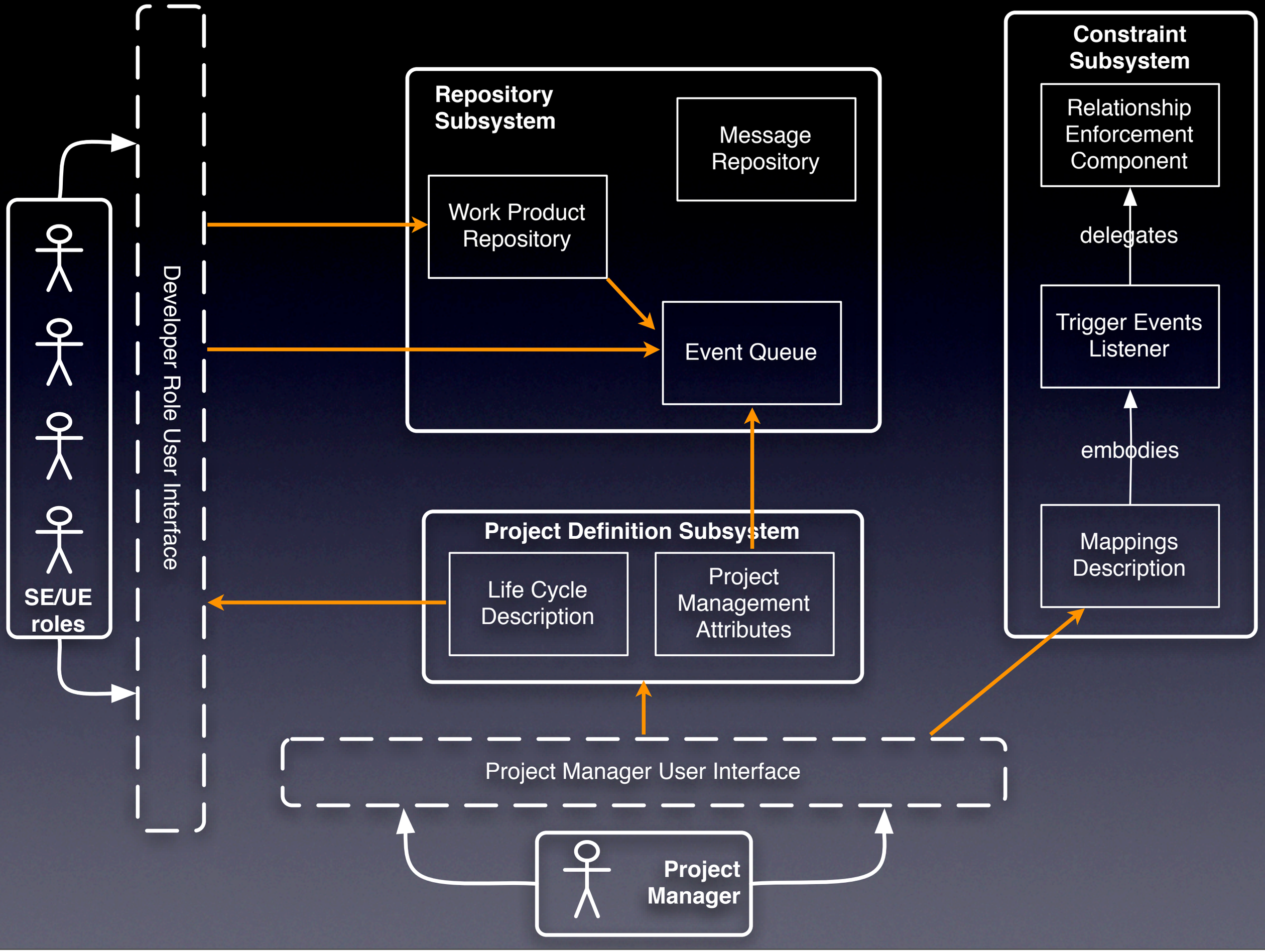

Repository Subsystem
Message Repository
Work Product Repository
Event Queue
Constraint Subsystem
Relationship Enforcement Component
delegates
Trigger Events Listener
embodies
Mappings Description
Developer Role User Interface
SE/UE roles
Project Definition Subsystem
Life Cycle Description
Project Management Attributes
Project Manager User Interface
Project Manager

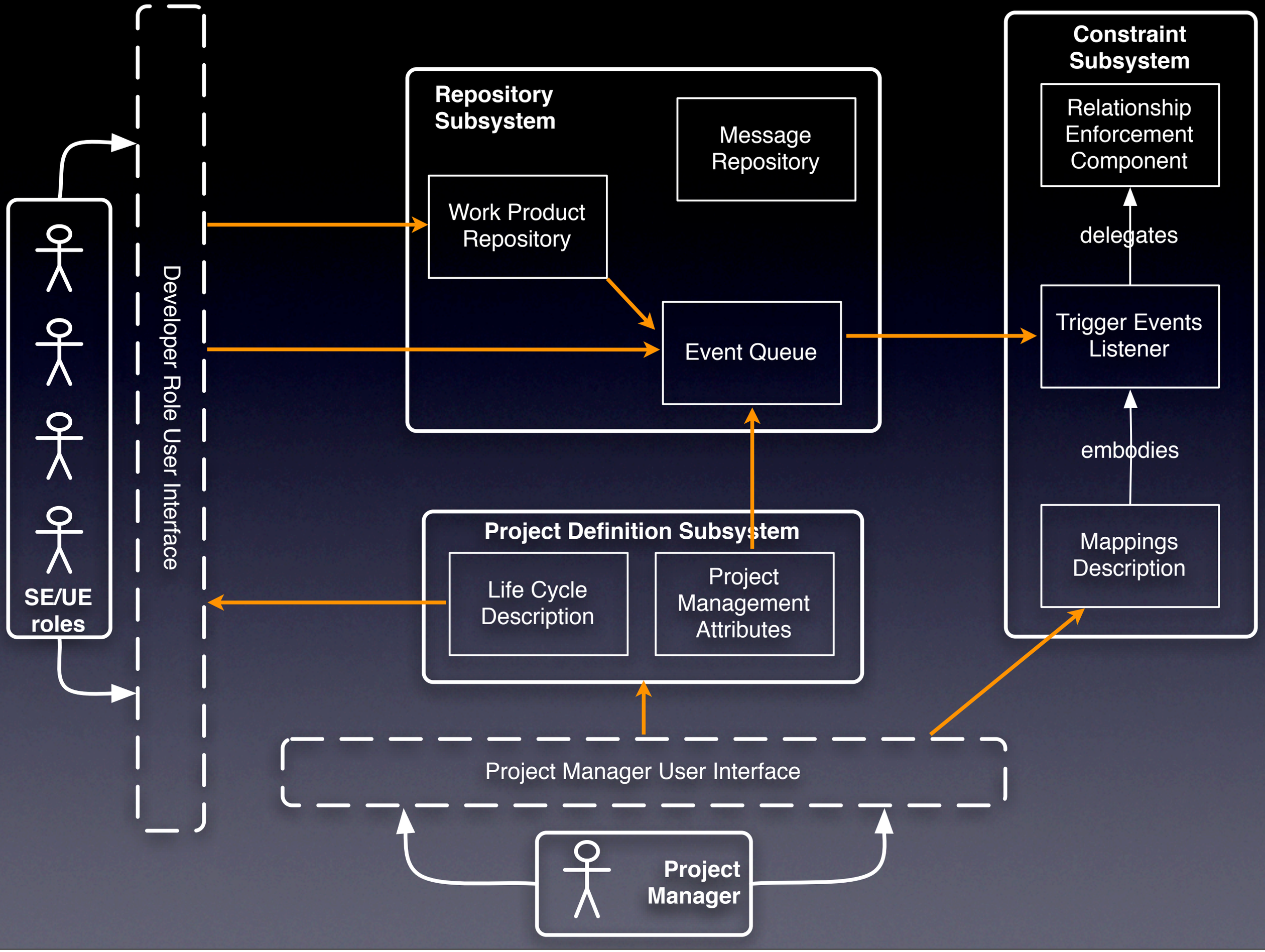

Repository Subsystem
Message Repository
Work Product Repository
Event Queue
Constraint Subsystem
Relationship Enforcement Component
delegates
Trigger Events Listener
embodies
Mappings Description
Developer Role User Interface
SE/UE roles
Project Definition Subsystem
Life Cycle Description
Project Management Attributes
Project Manager User Interface
Project Manager

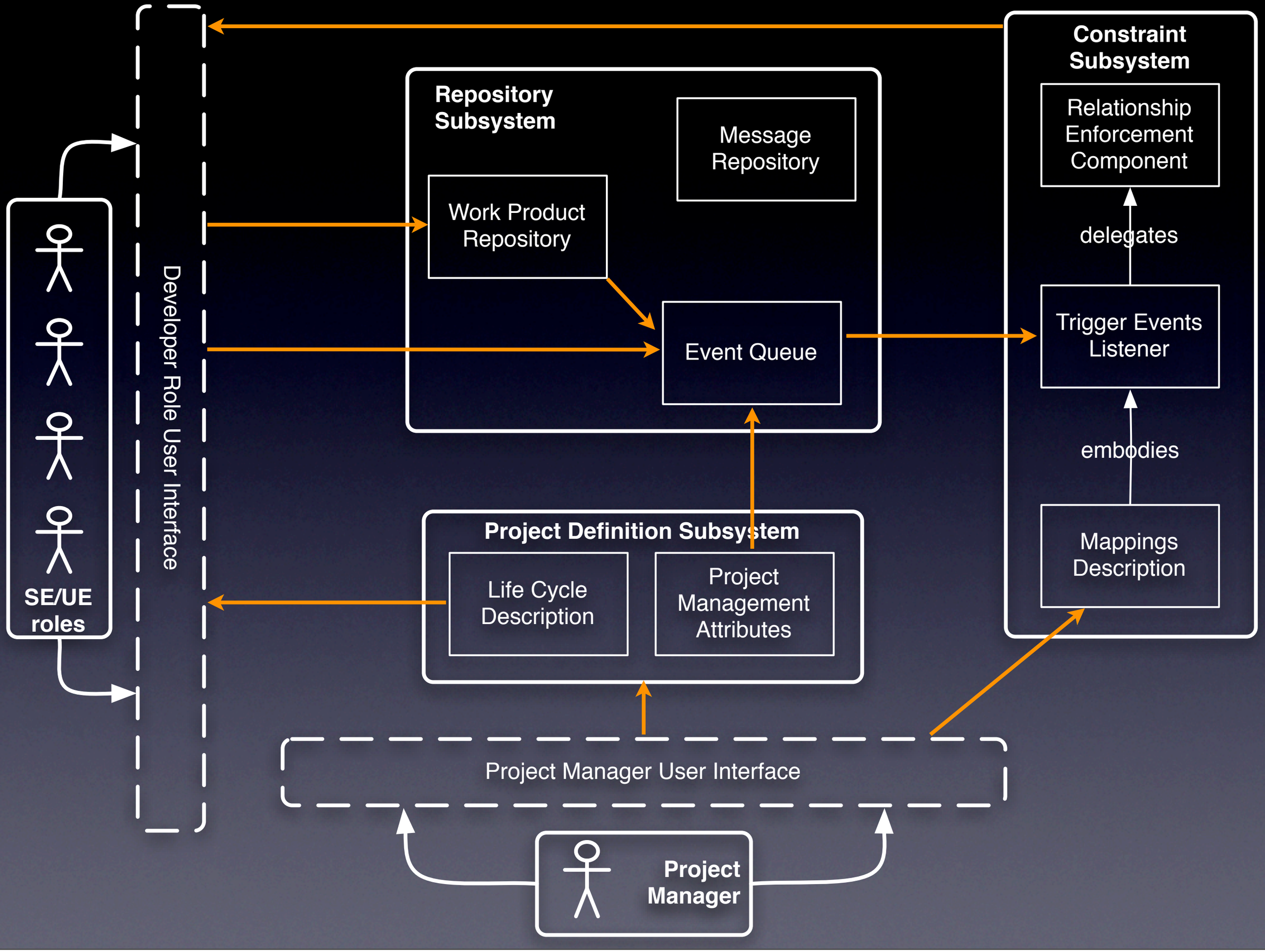

Repository Subsystem
Message Repository
Work Product Repository
Event Queue
Constraint Subsystem
Relationship Enforcement Component
delegates
Trigger Events Listener
embodies
Mappings Description
Developer Role User Interface
SE/UE roles
Project Definition Subsystem
Life Cycle Description
Project Management Attributes
Project Manager User Interface
Project Manager


Repository Subsystem
Message Repository
Work Product Repository
Event Queue
Constraint Subsystem
Relationship Enforcement Component
delegates
Trigger Events Listener
embodies
Mappings Description
Developer Role User Interface
SE/UE roles
Project Definition Subsystem
Life Cycle Description
Project Management Attributes
Project Manager User Interface
Project Manager


Repository Subsystem
Message Repository
Work Product Repository
Event Queue
Constraint Subsystem
Relationship Enforcement Component
delegates
Trigger Events Listener
embodies
Mappings Description
Developer Role User Interface
SE/UE roles
Project Definition Subsystem
Life Cycle Description
Project Management Attributes
Project Manager User Interface
Project Manager


Repository Subsystem
Message Repository
Work Product Repository
Event Queue
Developer Role User Interface
SE/UE roles
Project Definition Subsystem
Life Cycle Description
Project Management Attributes
Constraint Subsystem
Relationship Enforcement Component
delegates
Trigger Events Listener
embodies
Mappings Description
Project Manager User Interface
Project Manager


Repository Subsystem
Message Repository
Work Product Repository
Event Queue
Constraint Subsystem
Relationship Enforcement Component
delegates
Trigger Events Listener
embodies
Mappings Description
Developer Role User Interface
SE/UE roles
Project Definition Subsystem
Life Cycle Description
Project Management Attributes
Project Manager User Interface
Project Manager


Constraint Subsystem
Relationship Enforcement Component
delegates
Trigger Events Listener
embodies
Mappings Description
Repository Subsystem
Message Repository
Work Product Repository
Event Queue
Developer Role User Interface
SE/UE roles
Project Definition Subsystem
Life Cycle Description
Project Management Attributes
Project Manager User Interface
Project Manager

Ripple Implementation Instance

- Project definition subsystem
  - class schedule and project specifications
- Ripple constraint subsystem
  - wizard-of-oz
  - emails
- Work product repository
  - Google Groups email archive

Evaluation

- Systems-level investigation
  - full summative study not feasible
  - exploratory in nature (not all factors known yet)
  - too few teams per condition for complete comparison
  - unequal number of teams in each condition (1-3)
  - lack of control on balancing some teams

Evaluation Objectives

- Explore impact of communication factors
  - effectiveness of Ripple Implementation Framework
- Explore impact of cross-pollinated courses
  - effectiveness of joint SE-UE course

Key Assumption

Communication ≈ Ripple framework

Study Structure

Study Structure

CS 5704 - Software Engineering

- trained for software engineer role
- developed functional core and UI

Study Structure

CS 5704 - Software Engineering

- trained for software engineer role
- developed functional core and UI

CS 5714 - Usability Engineering

- trained for usability engineer role
- developed user interface prototypes

Study Structure

CS 5704 - Software Engineering

- trained for software engineer role
- developed functional core and UI

CS 5714 - Usability Engineering

- trained for usability engineer role
- developed user interface prototypes

A Teams
3 SE 2 3-4 UE

B Teams
3 SE 2 4 UE

C Team
3 SE/UE

D Teams
3
4-5 UE

Team Types -1

- A teams (ripple)
  - distinct SE/UE roles working together from the start
  - ideal real-world scenario
- B teams (non-ripple)
  - distinct SE/UE roles working independently until design
  - UE role brought in at the end of life cycle

Team Types -2

- C teams (dual-experts)
  - played both roles
  - software developers “doing” usability
- D teams (usability designers)
  - played UE role only
  - user experience contracted to external team
  - developed high-fidelity prototype without functional core

Overarching Hypotheses

H1. Communication is the key factor that affects quality of an interactive-software development effort

H2. Cross-pollinated SE-UE courses afford better learning experience

H1

C teams > A teams > B teams > D teams

- C teams: implicitly high amount of communication and situational awareness
- A teams: communication with counterpart roles + Ripple
- B teams: design ideas from two domains, albeit late
- D teams: narrow UI-only design focus

H2

- Joint SE-UE courses > SE- or UE-only courses
  - observe implications of communication
  - understand interactions with counterpart role
  - learn through experience and observation

Clients and Project

- Clients
  - team of 3 members from VT Horticulture Club
  - trained to minimize biases
  - interaction with teams controlled
- Project: software system for plant sale
  - online pre-ordering, point-of-sale, and inventory management systems

Study Instruments -1

Overall value index

Study Instruments -1

Overall value index

misc	welcome screen
	direction to plant sale
	make donation
browse	all
	group of letters (of name)
	categories
	last year's most famous

Study Instruments -1

Overall value index

		desirability
misc	welcome screen	high
	direction to plant sale	high
	make donation	low
browse	all	high
	group of letters (of name)	medium
	categories	high
	last year's most famous	low

Study Instruments -1

Overall value index

			team x		
		desirability	feature present	perceived quality	value index
misc	welcome screen	high			
	direction to plant sale	high			
	make donation	low			
browse	all	high			
	group of letters (of name)	medium			
	categories	high			
	last year's most famous	low			

Study Instruments -1

Overall value index

			team x		
		desirability	feature present	perceived quality	value index
misc	welcome screen	high	yes	fair	6
	direction to plant sale	high	no	-	0
	make donation	low	yes	good	3
browse	all	high	no	-	0
	group of letters (of name)	medium	yes	poor	2
	categories	high	yes	good	9
	last year's most famous	low	no	-	0

high	medium	low
3	2	1

yes	no
1	0

good	fair	poor
3	2	1

Study Instruments -2

- Structured journals
  - individual/group hours
  - unexpected changes
  - issues with SE-UE interaction
- Email archives
  - unit of communication
  - tagged with keywords
  - frequency counts

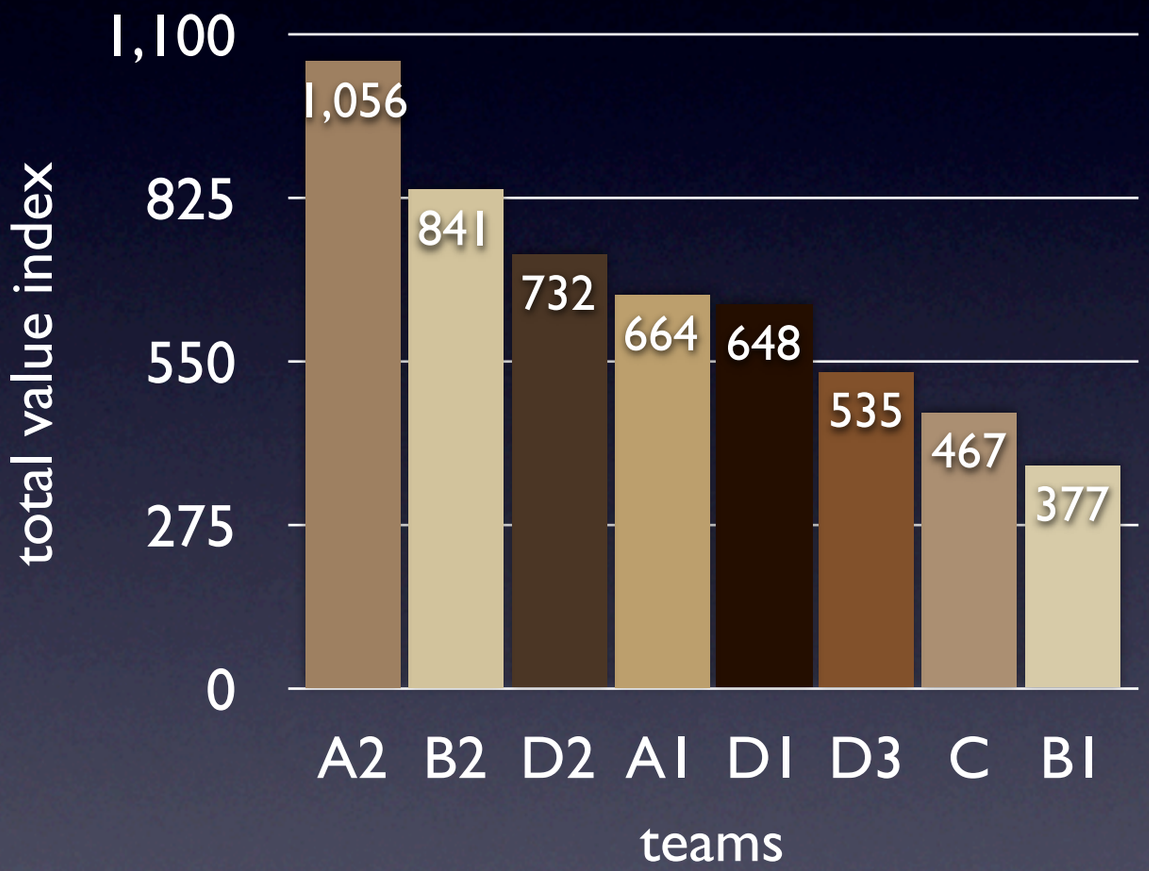
Study Instruments -3

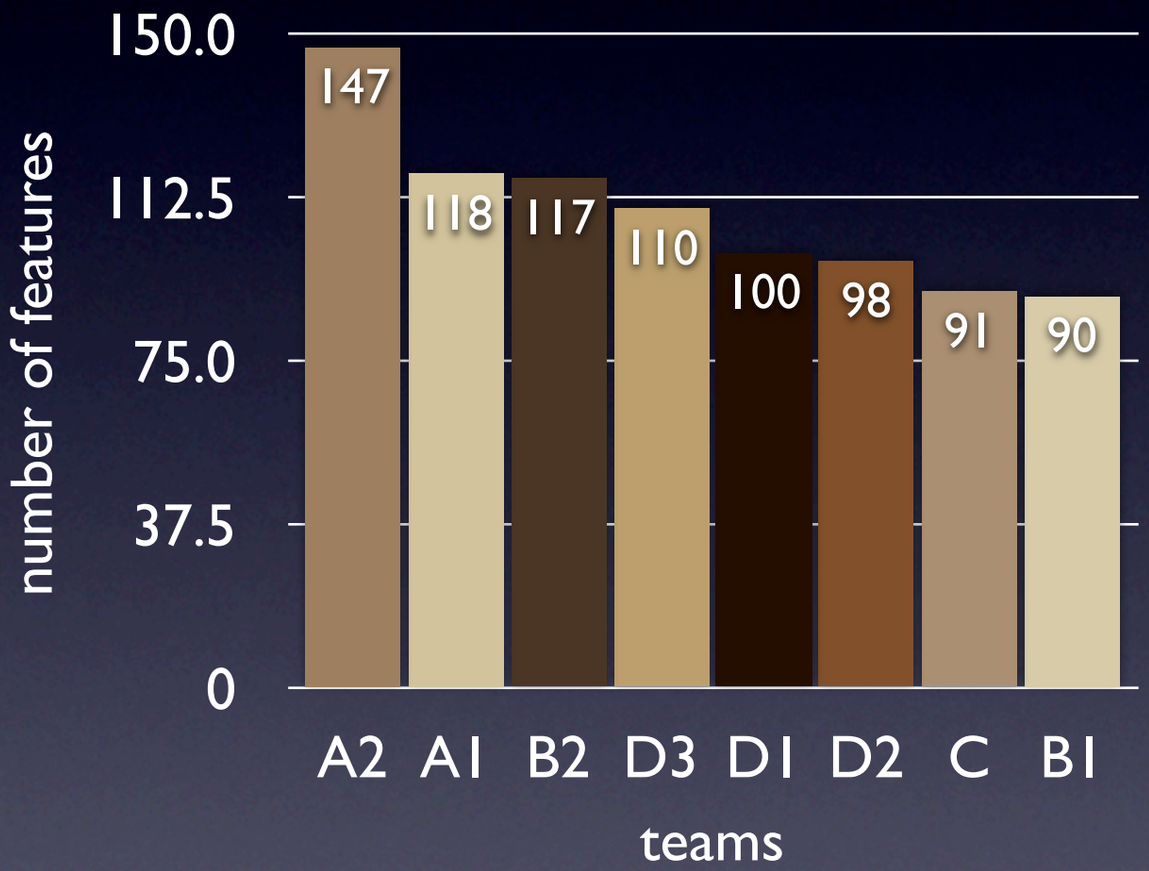
- End-of-semester symposium
  - process, interaction, what worked, did not, heuristics for real-world developers
- Group interviews
- Surveys (2)
- Client feedback
  - after each meeting
  - after overall product comparison

Investigative Approach

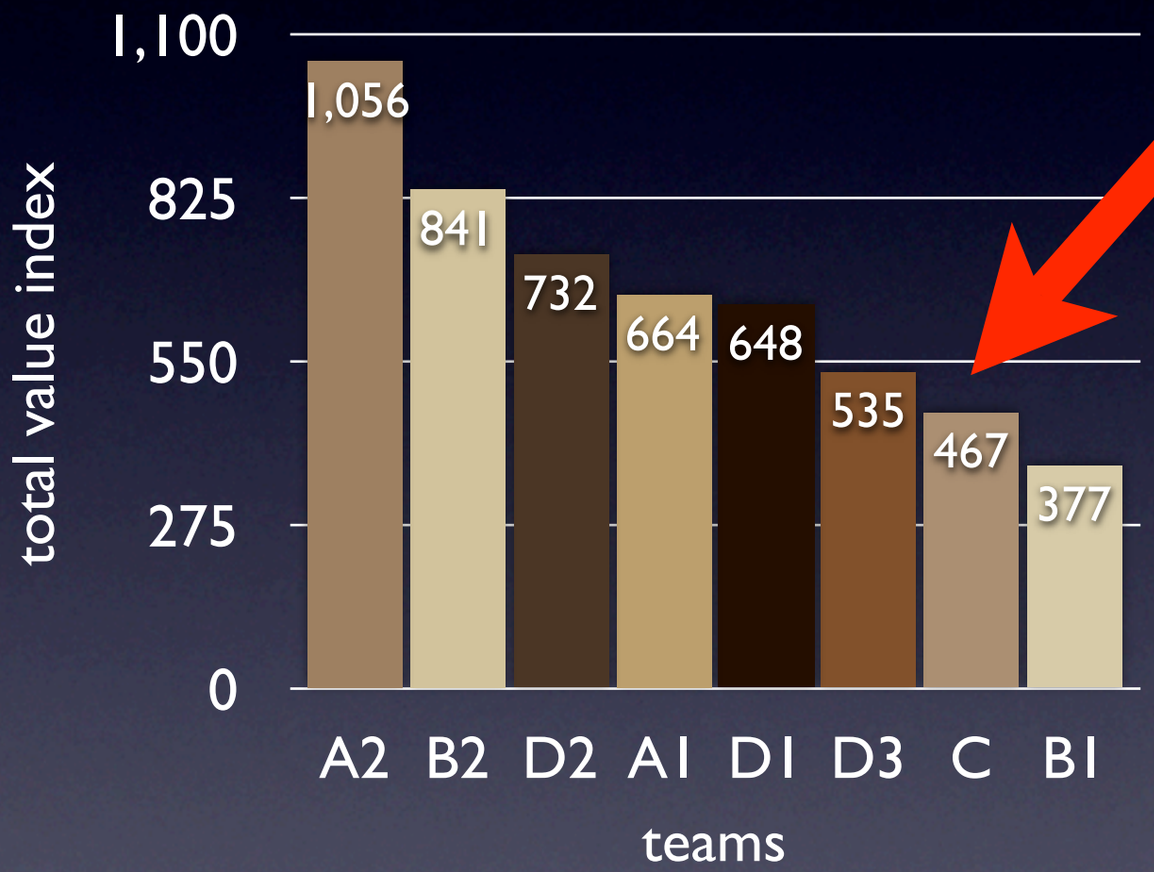
- No single instrument describes the complete “picture”
- Analogy of detective work

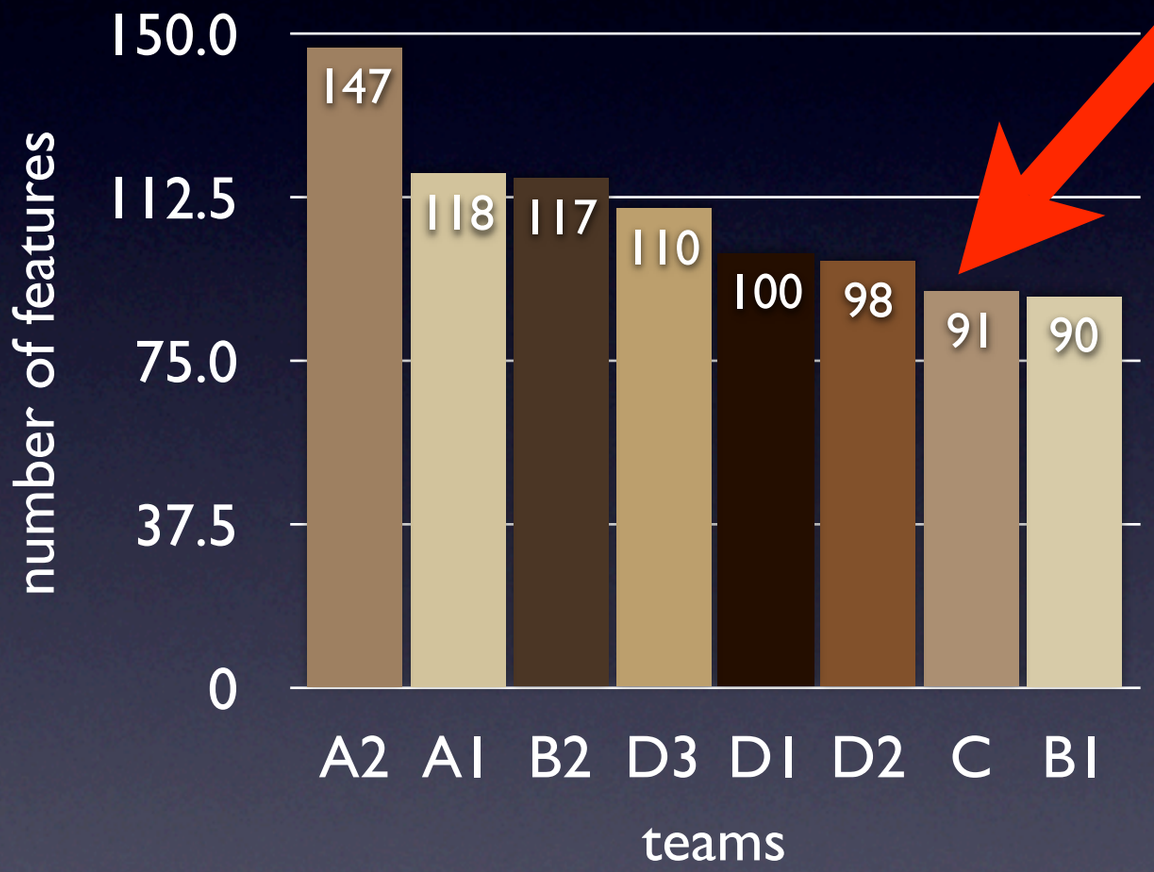
Observations





Observations





C > A [not verified]

- Inherent conflict of interest

  "We couldn't be like the other teams where they said, what the heck it is not our problem... it is the problem of the SE team. We were the SE team as well!"

- Life cycle constraints

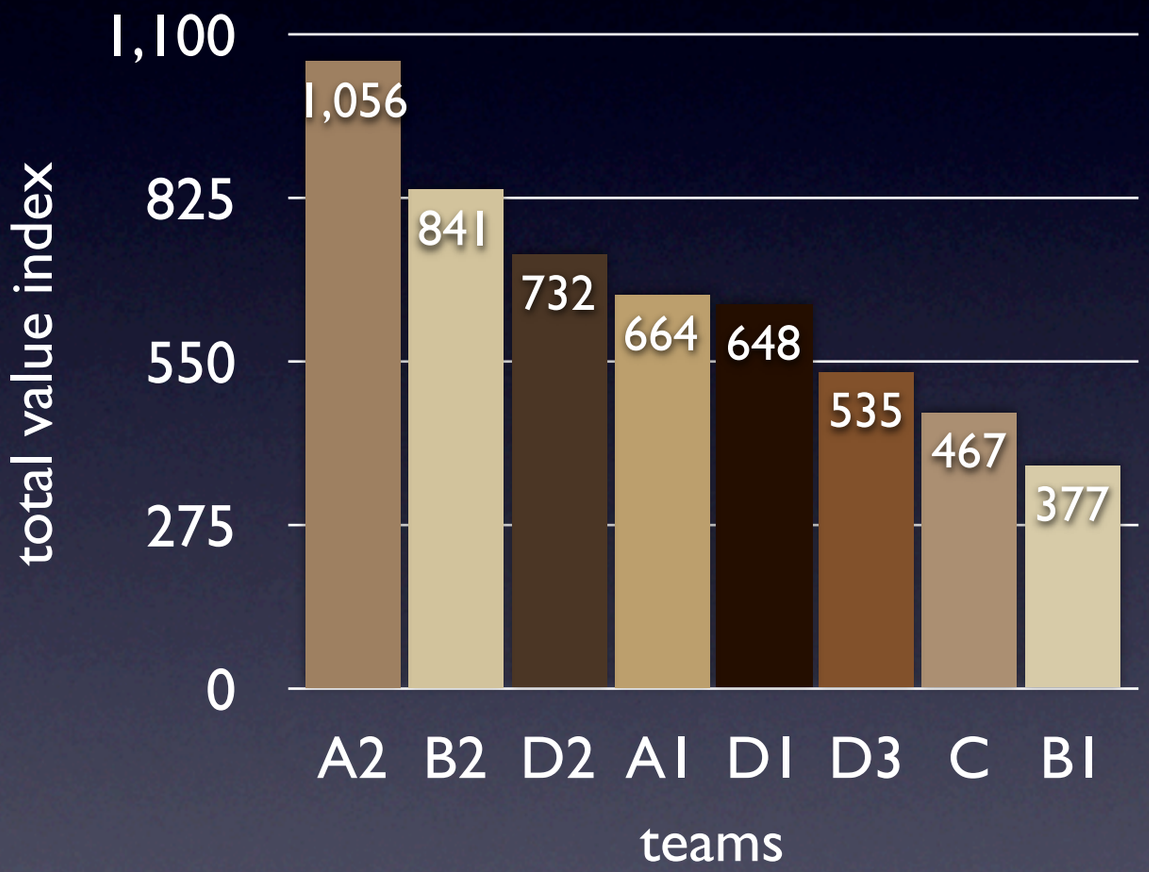
+ Overlap of work activities, reduced work load

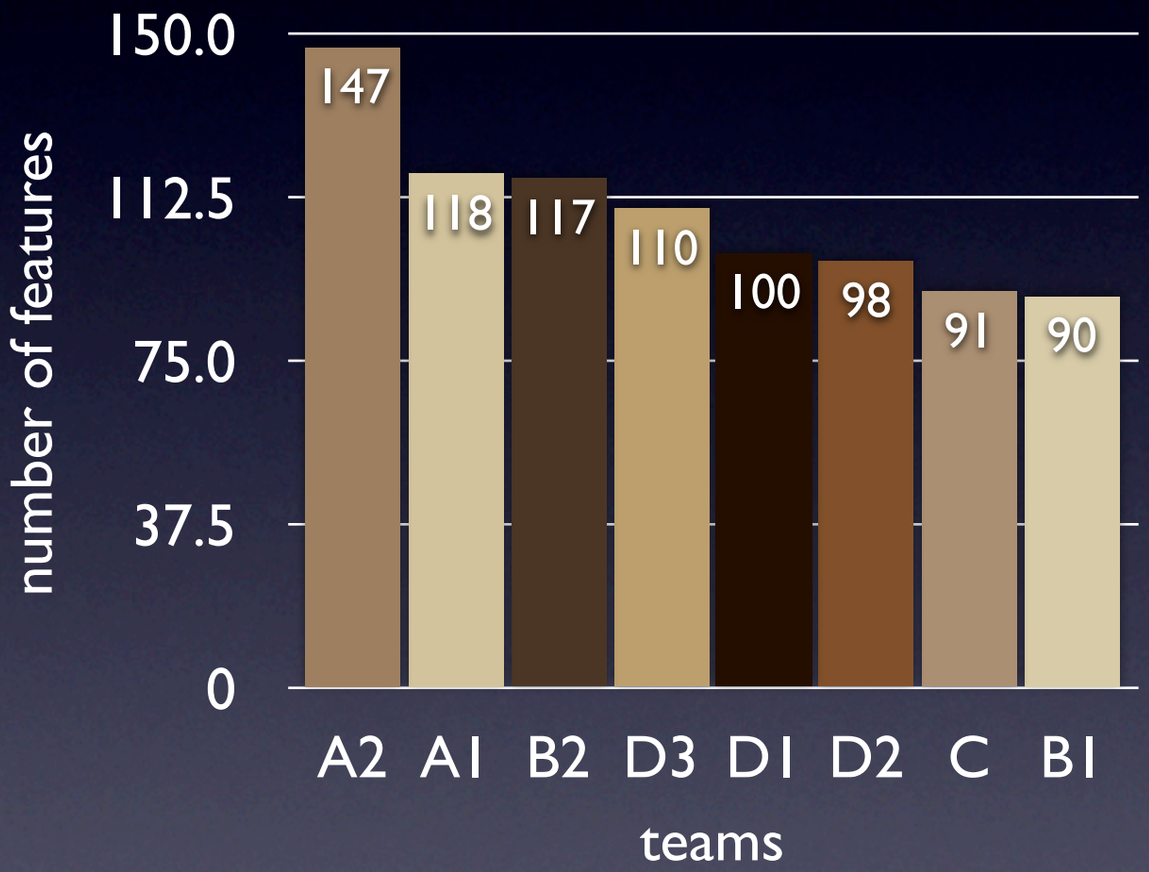
+ Immediate collaboration

- Reassurance of being on the right track

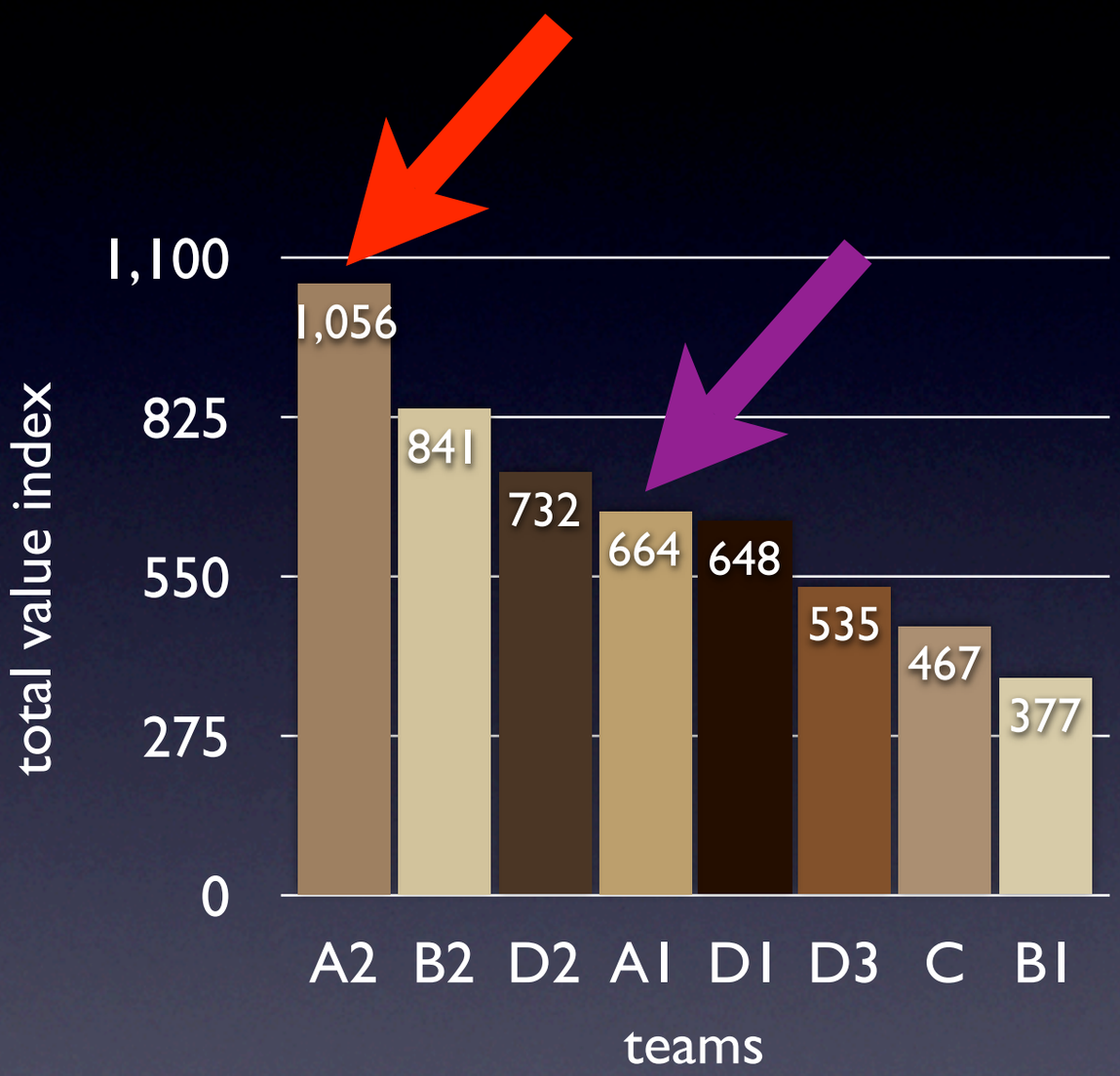
Clients: *"The first screen was nice but the rest is unprofessional. No imagination and not creative. Had stuff with a box around it. Very minimal"*

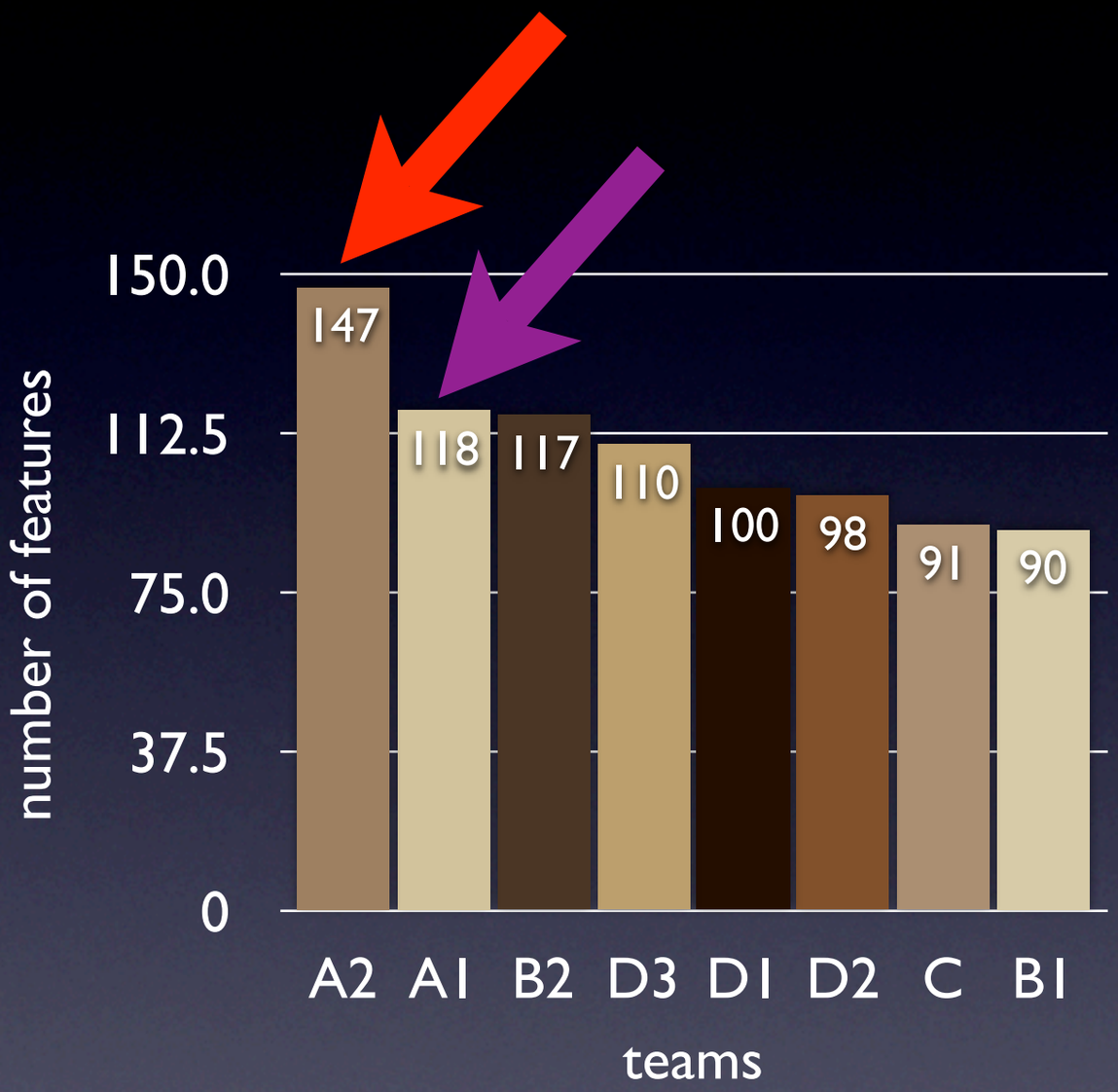
Observations





Observations





A > B [partly verified]

- Team A2 - performed as hypothesized
- Good communication and collaboration
- Periodic synchronization

  "*we also had several meetings outside the ones that just [Ripple required us]. And for any other meetings that say the UE team had, we would always make sure we had at least one SE representative to report back to the rest of the group. So […] everyone generally knew where the other team was*"

- Client: "*They are amazing! Wow*"

Team A1 - Disaster Team

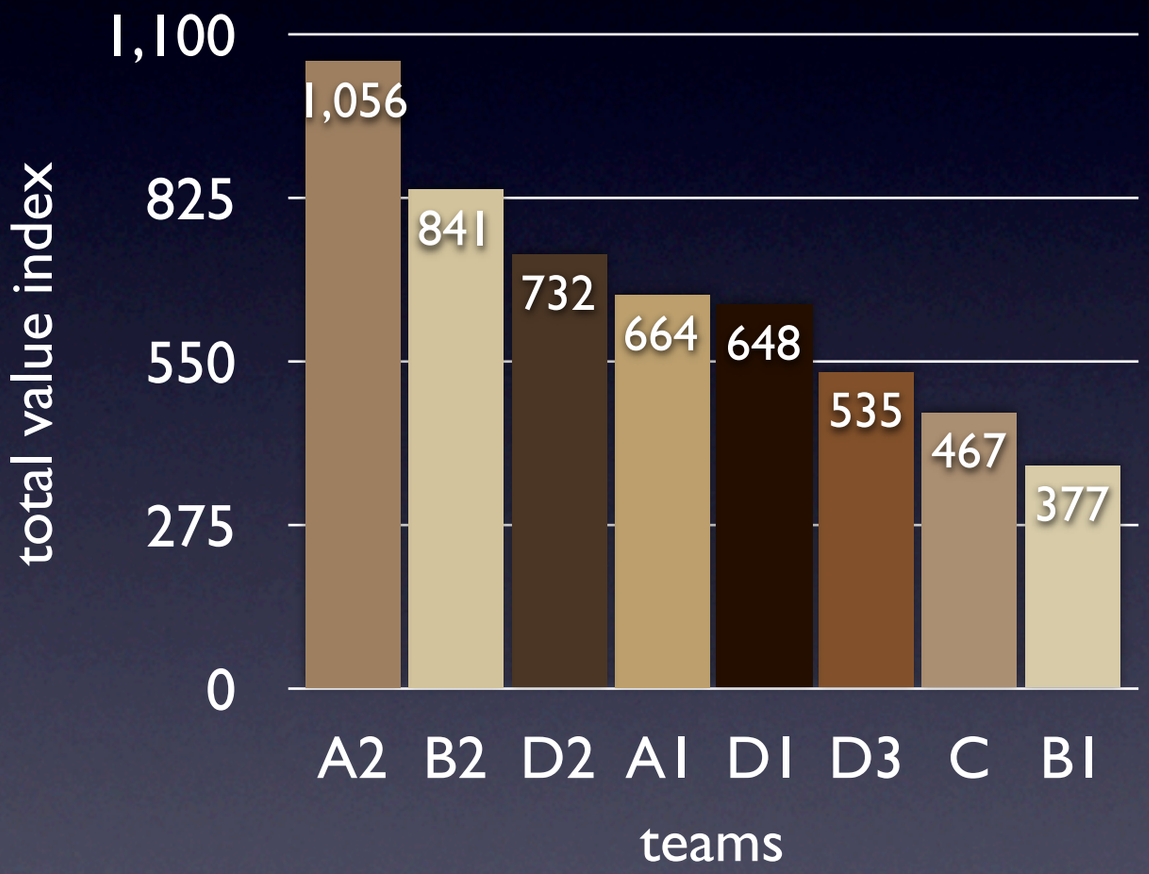
- Incompatible UI and backend components

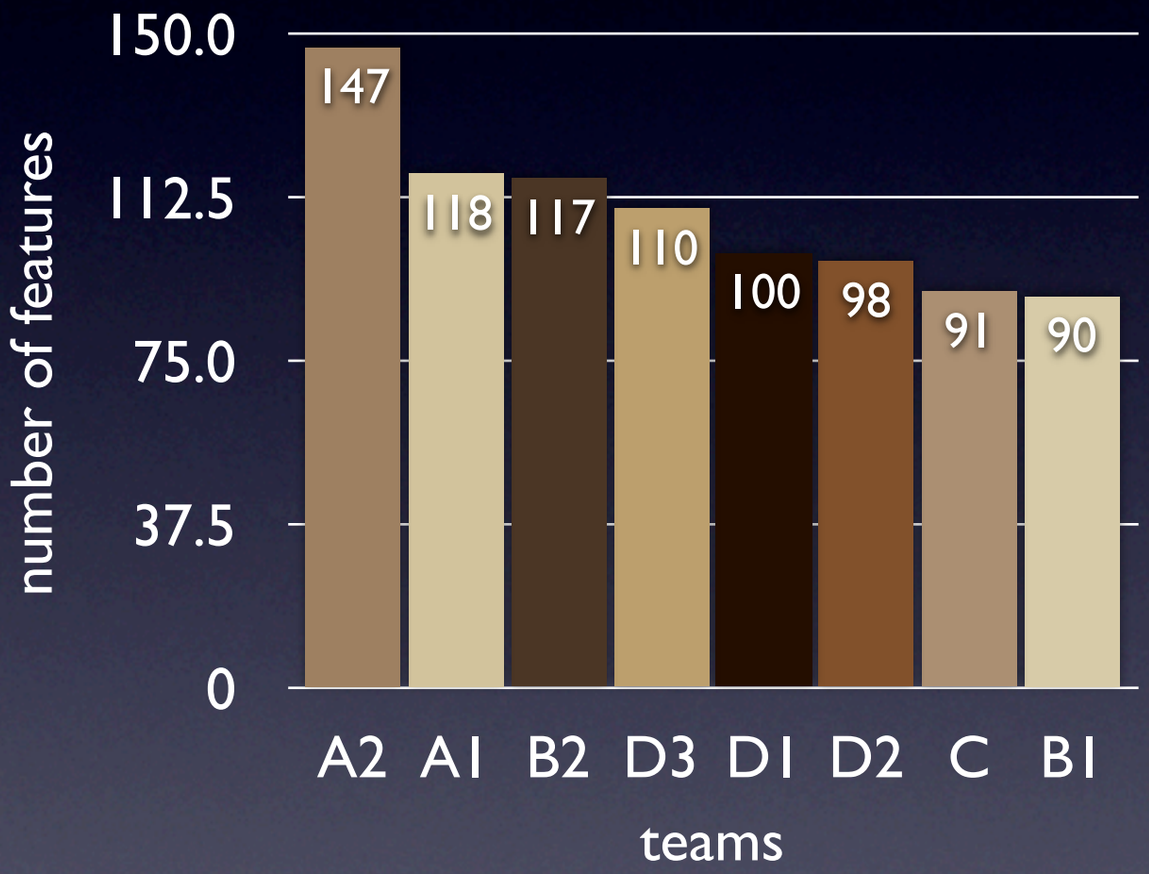
  "*the first thing that we found was that we had different priorities from the beginning.[…] This led to a totally different idea of what shopping is! To the UE team shopping was, you have time to kill, you were looking for some plants, you are browsing around, but for the SE team it was like, a serial killer is chasing you and you had to buy a rose plant. So surprise, our designs were nothing alike*"

- Uncooperative self-proclaimed SE lead
- Competent team that failed (needed intervention)

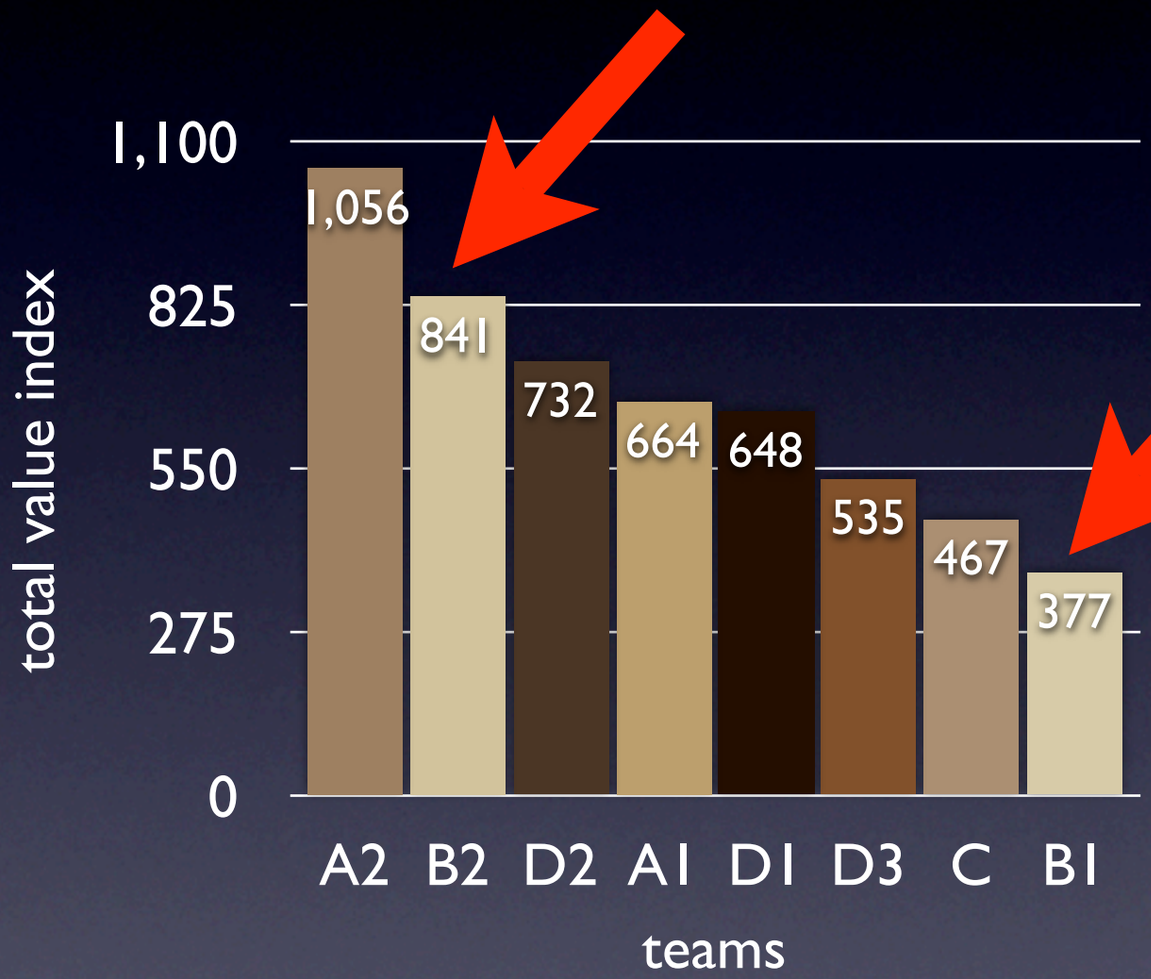
"We (at least I) have not really been paying attention to the UE guys since their input does not really affect our current progress. I'm 99% sure I will just end up scrapping their UI and writing my own since I want to use AJAX and I don't think they even know what that means. AJAX is very difficult to implement if your UI is not designed for it, as all your content is dynamic. I consider myself primarily an HCI person, so I suppose I think I can do a competent design job on the UI (I like to do my own design work). [I] haven't really brought this up yet [with the UE role and] what they don't know won't hurt them."

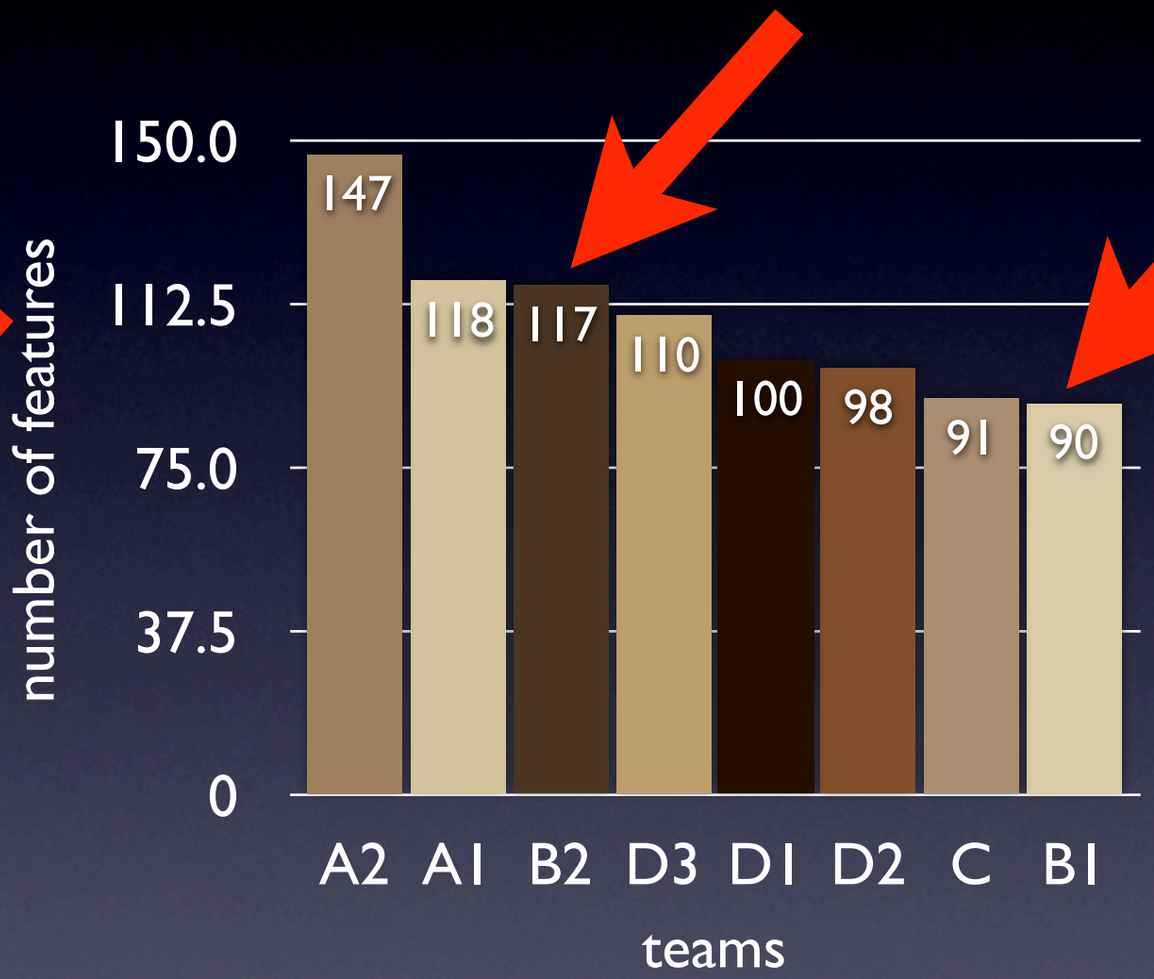
Observations





Observations





B > D [not verified]

- Team B2 - Second out of eight
  - Power team with prior shared-experience
  - Adaptive to new technologies, forward looking
  - Reuse of UE's prototype code in SE modules
  - Negotiation, role distinction, respect for counterpart role
  - One of clients' functional favorites

"I very early on told myself that I am not going to question them at all on SE process, which I did not. Even though I wanted to come out and say 'what is it you guys are doing?' I very early on said that I am a UE guy and [will] keep [to] my thing"

Team B1 - As Hypothesized

- Challenges due to communication constraint
- Interaction inertia
- Inability to resolve differences in scope and vision

“One of the weirdest things for us was the scope of the design, we came into our first combined meeting expecting to only having to do a point-of-sale, inventory, the standard stuff we talked about all the semester, then they told us they wanted to rewrite the entire site as well as having a news site events, and a whole host of different things!"

B1 - Territoriality Issues

"I, along with my UE team members, were very very very upset! And then to add on to that, she wasn't even at the meeting today to explain herself!!!!! […] Also, it was not her job to change anything on the front end!!! As part of the SE team, it is her job to only mess with the backend!!! I'm so mad that she did that. I liked the way we had it before and it shouldn't have been changed, especially without asking anyone else or even notifying someone about it!!!"

H1 Revised

- Assumptions corrected
  - communication is critical, but
    - preempted by social/organizational factors
    - futility of ripple when ignored
- Success defined by people who have to utilize communication

B > D [not verified]

- D teams - performed better than hypothesis
  - constraint-free environment
  - creative prototypes with broad designs
- Appropriate for exploratory systems
- Potential practical problems in real world

Clients on team D2: "*This is the coolest design. They had a very broad design, with a lot of extra features such as zone map, etc. Their 'add to cart' is the best. The information is very detailed. Love it. This is like the real thing. Wow. Unbelievable*"

Joint SE-UE Better [verified]

- Joint SE-UE more valuable than individual ones
- Better learning experience
  - by condition (A > D)
- Student: “most valuable class [...] at Tech”
- Real clients provide better learning experience
- Laboratory approach to teaching a great success

Relevant Contributions

- Pioneered a methodology to address a long-standing problem of connecting SE and UE
  - Created a framework to foster and structure communication
  - Demonstrated utility using an exploratory study
- Serious implications on quality of systems
  - Need for checks and balances to ensure positive team dynamics
  - Communication impacts on geographically distributed teams
- Demonstrated effectiveness of cross-pollinated courses

end