

Thorny Issues in Software Requirements

Sponsored by:



Karl Wieggers

Principal Consultant, Process Impact
www.processimpact.com



Sponsor: iRise

- Interactive
- Collaborative
- Integrated
- Scalable
- Successful

iRise is transforming the way businesses around the world define and build software.

From simulations to testing to functional specs to code generation, successful teams use iRise as the central communication platform to drive all phases of the software development lifecycle and requirements management - from concept to launch.

Source Book



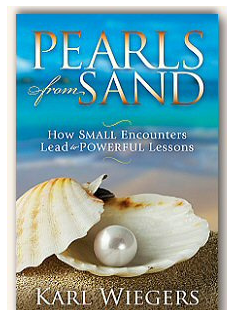
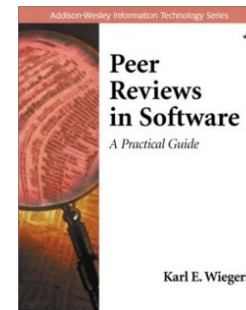
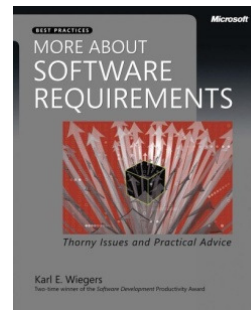
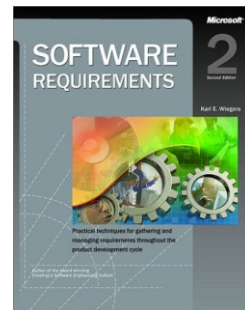
- ❑ *More About Software Requirements*, by Karl E. Wiegerts (Microsoft Press, 2006)

Featured Speaker



Karl Wiegiers

Principal Consultant, Process Impact
www.processimpact.com



Phone #:

503-698-9620

E-mail:

karl@processimpact.com

Blog:

Consulting Tips & Tricks Blog: www.karlconsulting.blogspot.com

How Much Detail Do You Need?

Less Detail

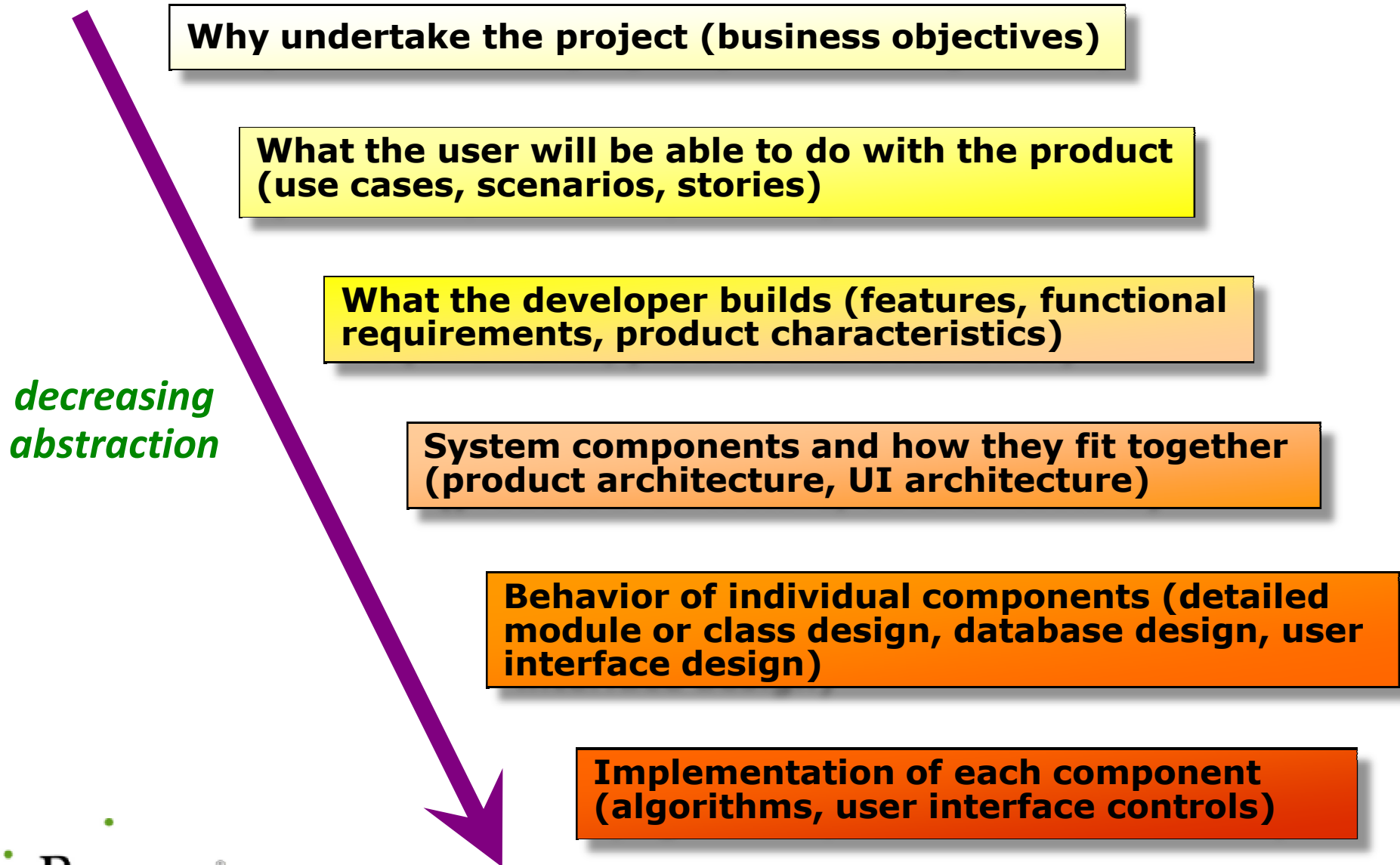
- Extensive customer involvement
- Developers have considerable domain experience
- Precedents are available
- Package solution will be used

**Amount of
detail needed
in
requirements**

More Detail

- Outsourced development
- Team is geographically dispersed
- Testing will be based on requirements
- Accurate estimates are needed
- Requirements traceability is needed

The Fuzzy Line: Requirements versus Design



Is It a Requirement or Is It Design?

“When the alarm system is armed and a sensor is triggered, the user shall be able to enter a numeric passcode to disarm the system.”

The system shall provide feedback as the user enters the passcode.

The system shall sound an audible tone after the user enters each digit of the passcode.

The system shall sound a tone of 1000 Hz at a volume of 75 dB for 0.5 seconds after the user enters each digit of the passcode.

The system shall contain an audio signal generator capable of producing constant tones ranging from 100-2000 Hz, inclusive, at a volume between 70 and 90 dB, inclusive.

The system shall contain a one-inch speaker with a frequency range of 80-8000 Hz capable of handling power input from 0.5 to 3 watts.

To Duplicate or Not To Duplicate

■ Benefits of duplicating information

- 📄 readers have all information they need in one place
- 📄 portions of work are encapsulated

■ Risks of duplicating information

- 📄 duplicated information might not be exactly identical
- 📄 changes can lead to inconsistencies

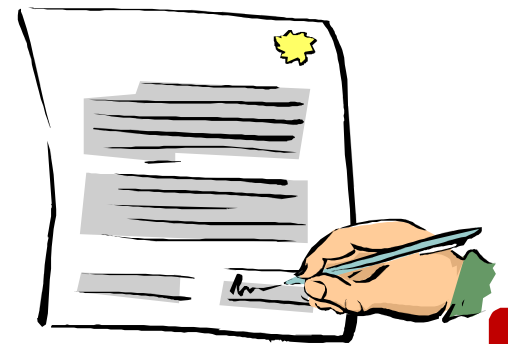
■ Techniques to avoid duplication

- 📄 cross-referencing in Word documents
- 📄 hyperlinks in documents
- 📄 traceability links
- 📄 requirements management tools



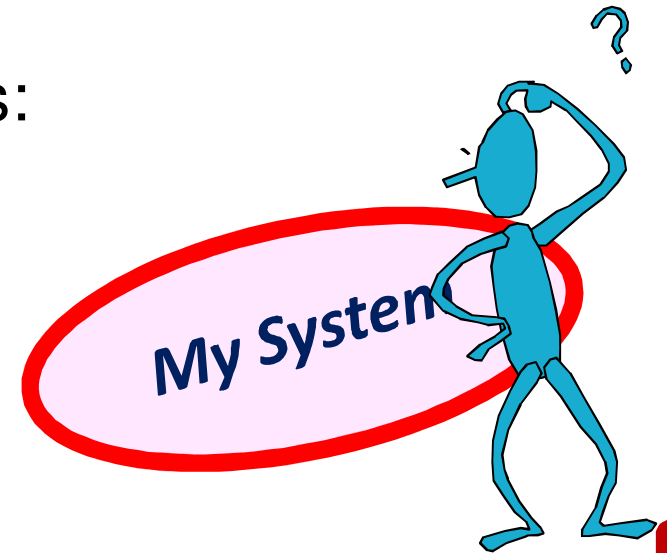
What Does “Sign-Off” Mean?

- Sometimes nothing!
- A suggestion:
 - 📄 “I agree that this document represents our best understanding of the requirements for this release today and that the system described will satisfy our needs as we currently understand them.
 - 📄 I agree to make future changes in this baseline through the project’s defined change process.
 - 📄 I realize that approved changes might require us to renegotiate the cost, resource, and schedule commitments of this project.”

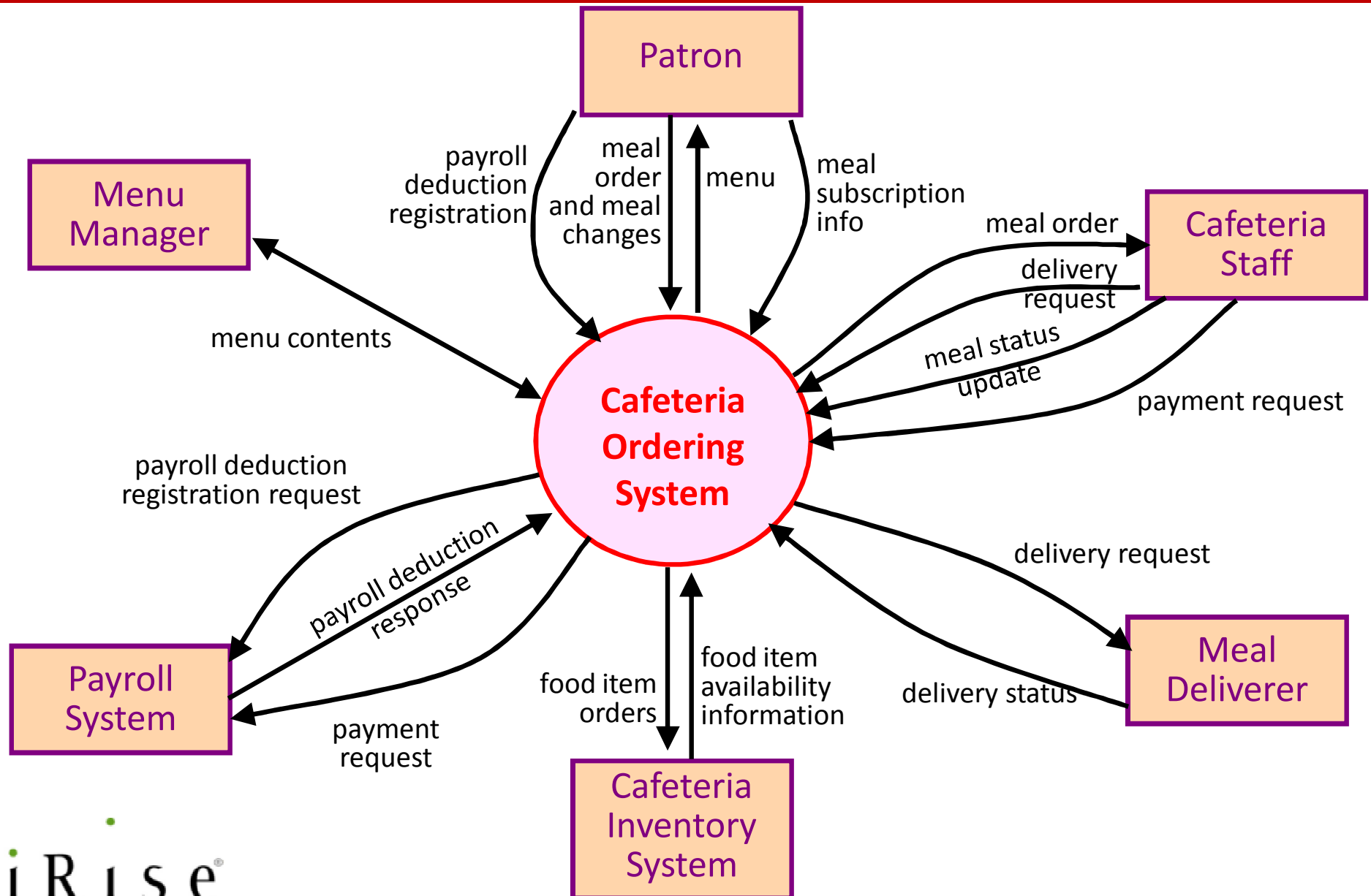


Depicting Project Scope

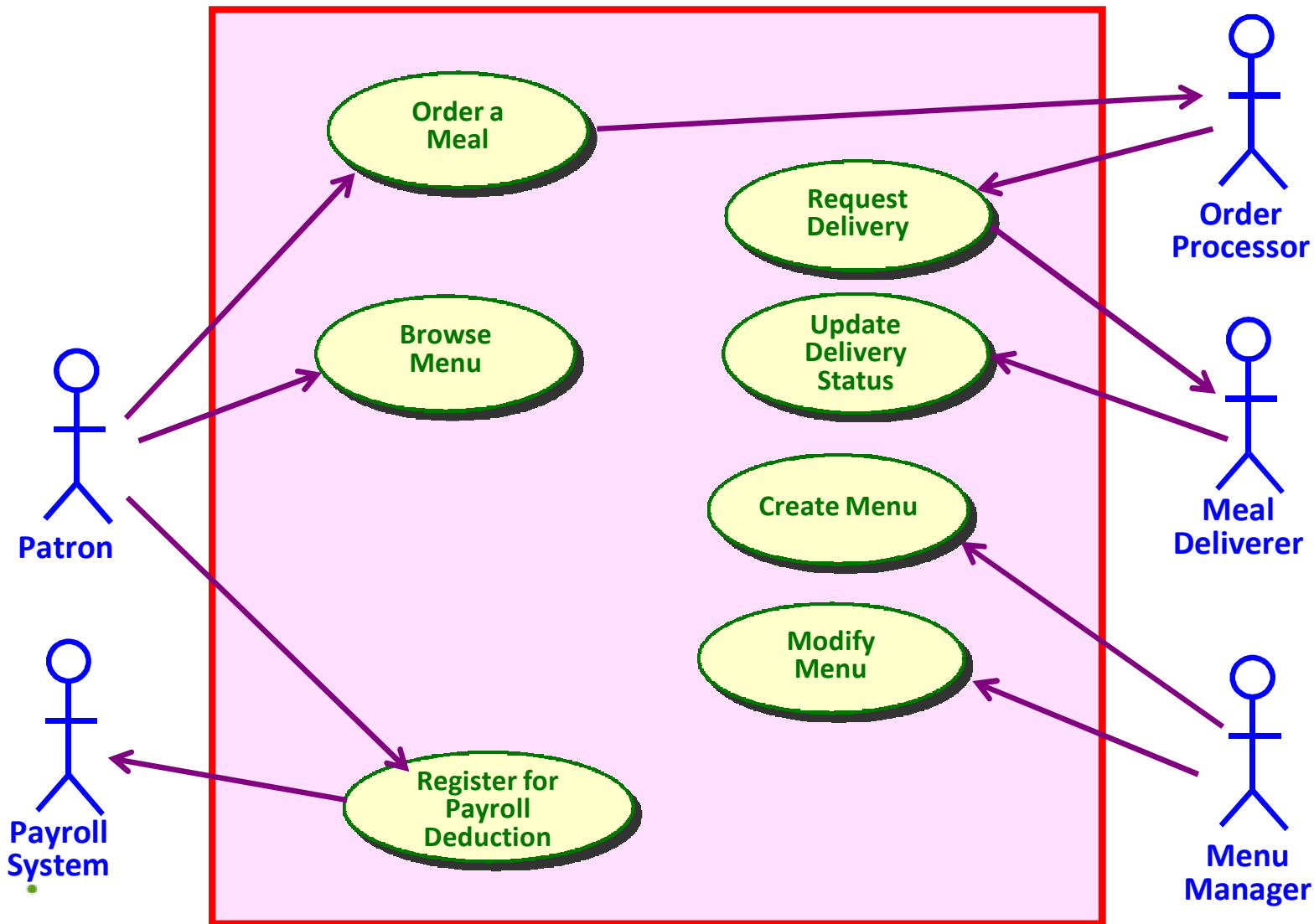
- Define what some portion of the product will and won't do
- Permit making realistic project commitments
- Enable prioritization and release planning
- Facilitate change decisions
- Manage scope creep
- Some scope representation techniques:
 - Context Diagram
 - Use Case Diagram
 - Feature Roadmap



Sample Context Diagram

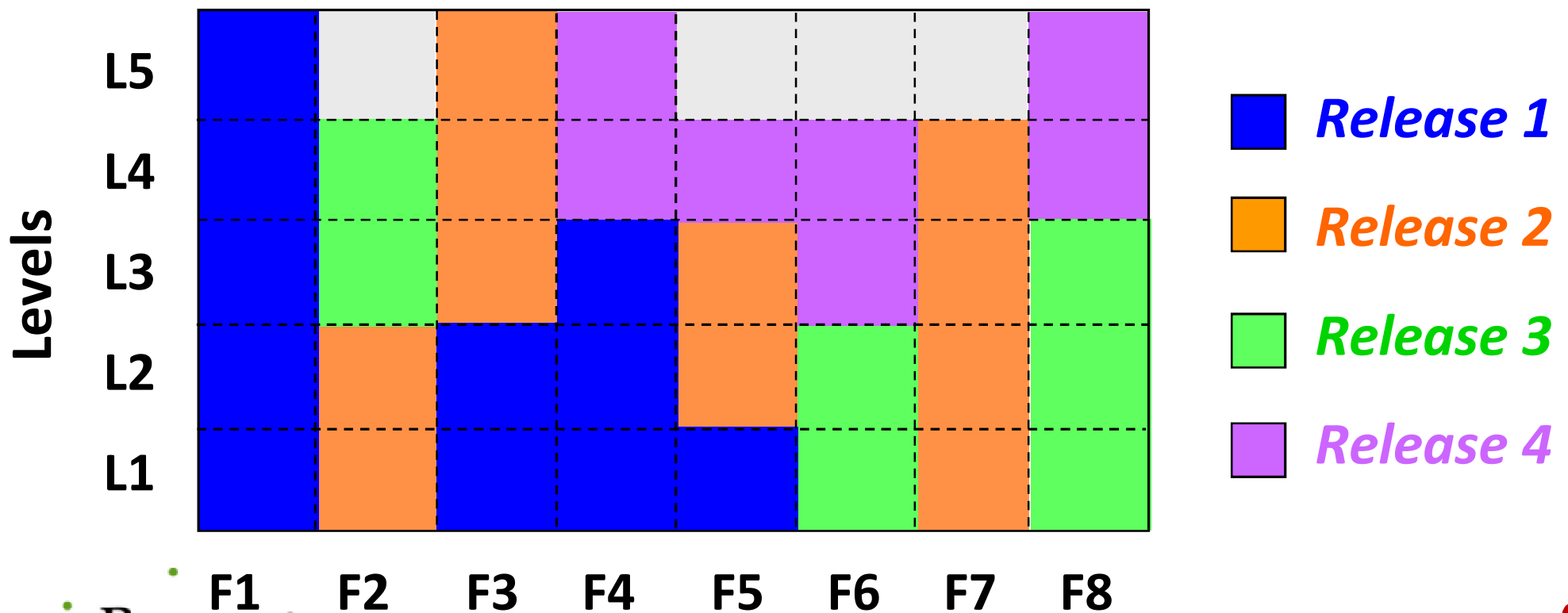


Sample Use Case Diagram



Feature Levels

- 8 features, 4 planned releases
- 4 or 5 enrichment “levels” for each feature.

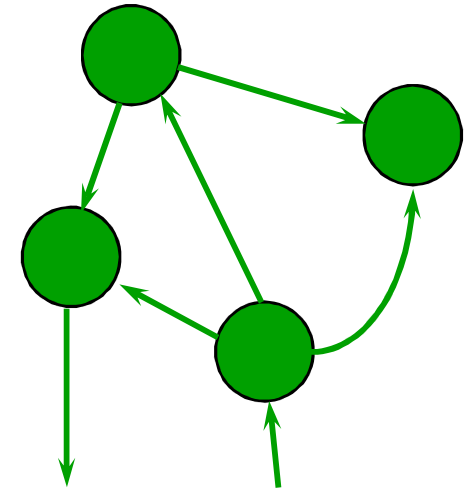


Sample Feature Roadmap

<i>Feature</i>	<i>Release 1</i>	<i>Release 2</i>	<i>Release 3</i>
Create cafeteria menus	Fully implemented		
Order meals from cafeteria	Standard individual meals from lunch menu only; payroll deduction only	Also accept orders for breakfasts and dinners; take credit and debit cards	Accept group meal orders for meetings and events
Order meals from restaurants	Not implemented	Not implemented	Fully implemented
Register for meal payment options	Register for payroll deduction payments only	Register for credit card and debit card payments	
Request meal delivery	Delivery only to company campus sites	Add delivery from cafeteria to certain off-site locations	Add delivery from restaurants to all delivery locations
Manage meal subscriptions	Implemented if time permits	Fully implemented	

Using Multiple Requirements Views

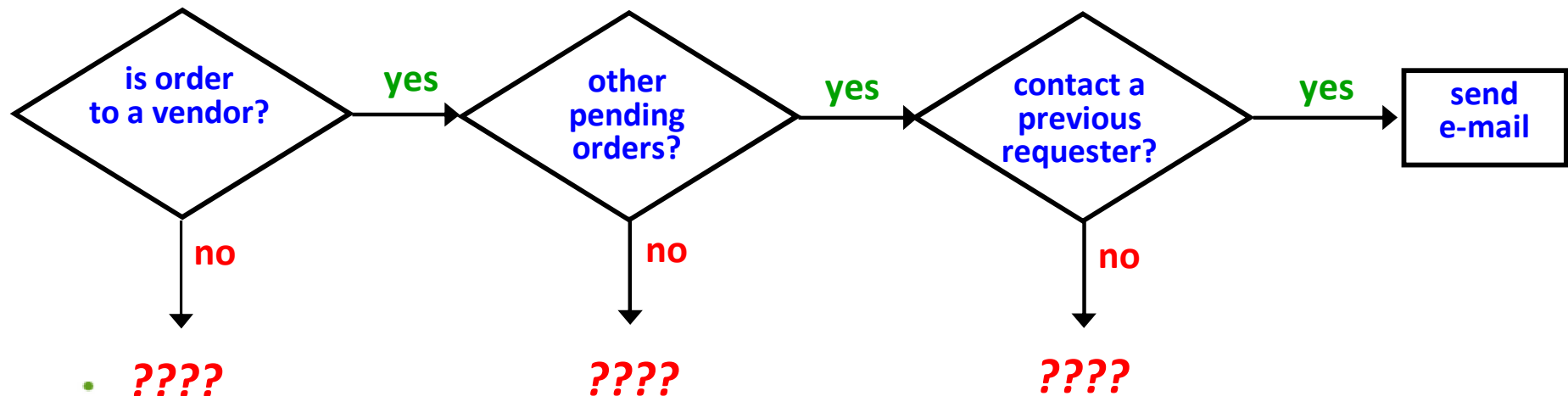
- Natural language text
- Graphical analysis models
- Decision tables and decision trees
- Test cases
- Prototypes and screen designs
- Tables and structured lists
- Mathematical expressions
- Data dictionary
- Photographs, video clips, audio clips



Example of Other Representation Techniques

Text: *If an order is placed for a chemical to a vendor, the system shall check to see if there are any other pending orders for that chemical. If there are, the system shall display the vendor name, vendor catalog number, and the name of the person who placed each previous order. If the user wishes to contact any person who placed a previous order, the system shall allow the user to send that person an e-mail message.*

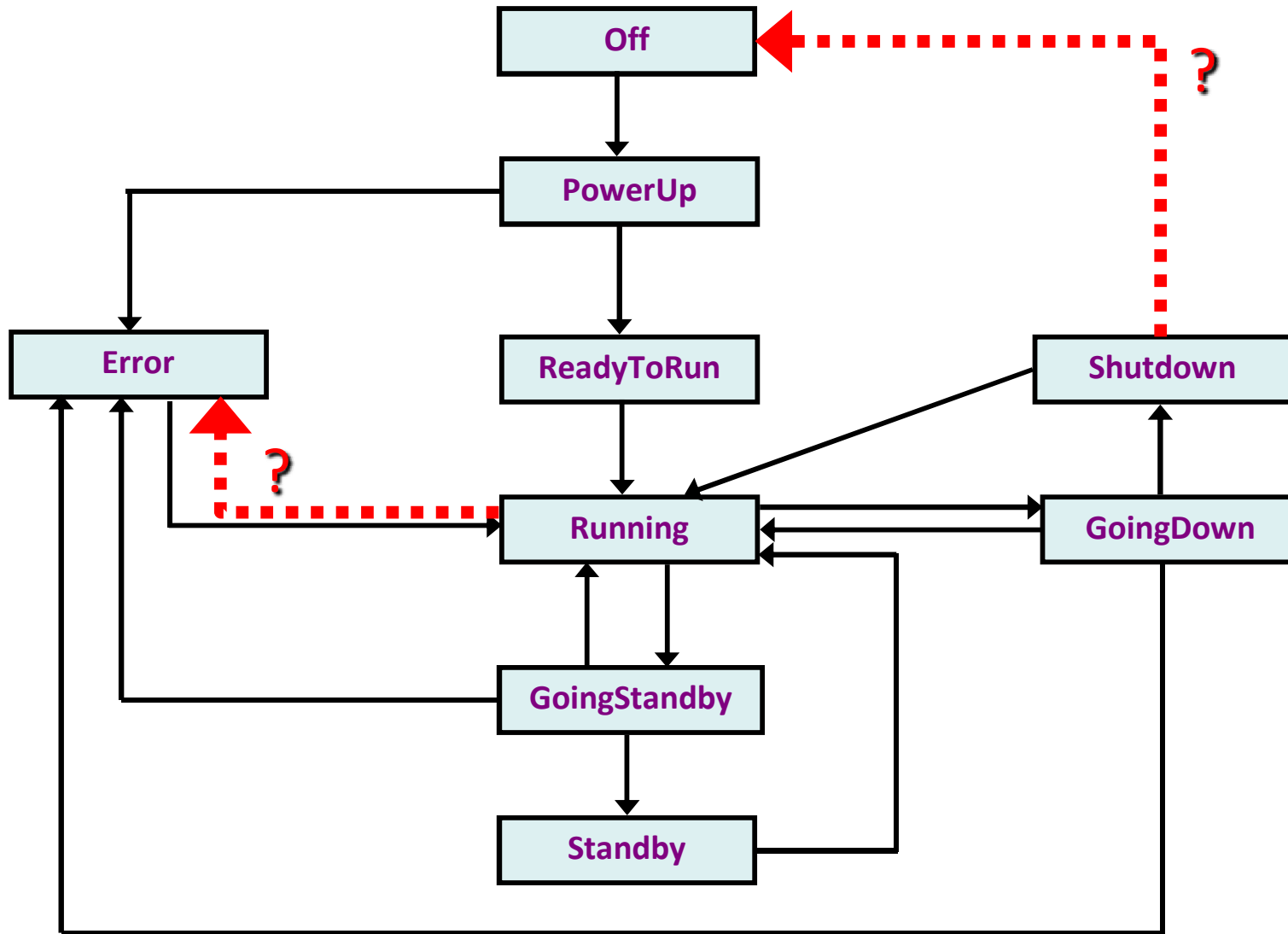
Decision Tree:



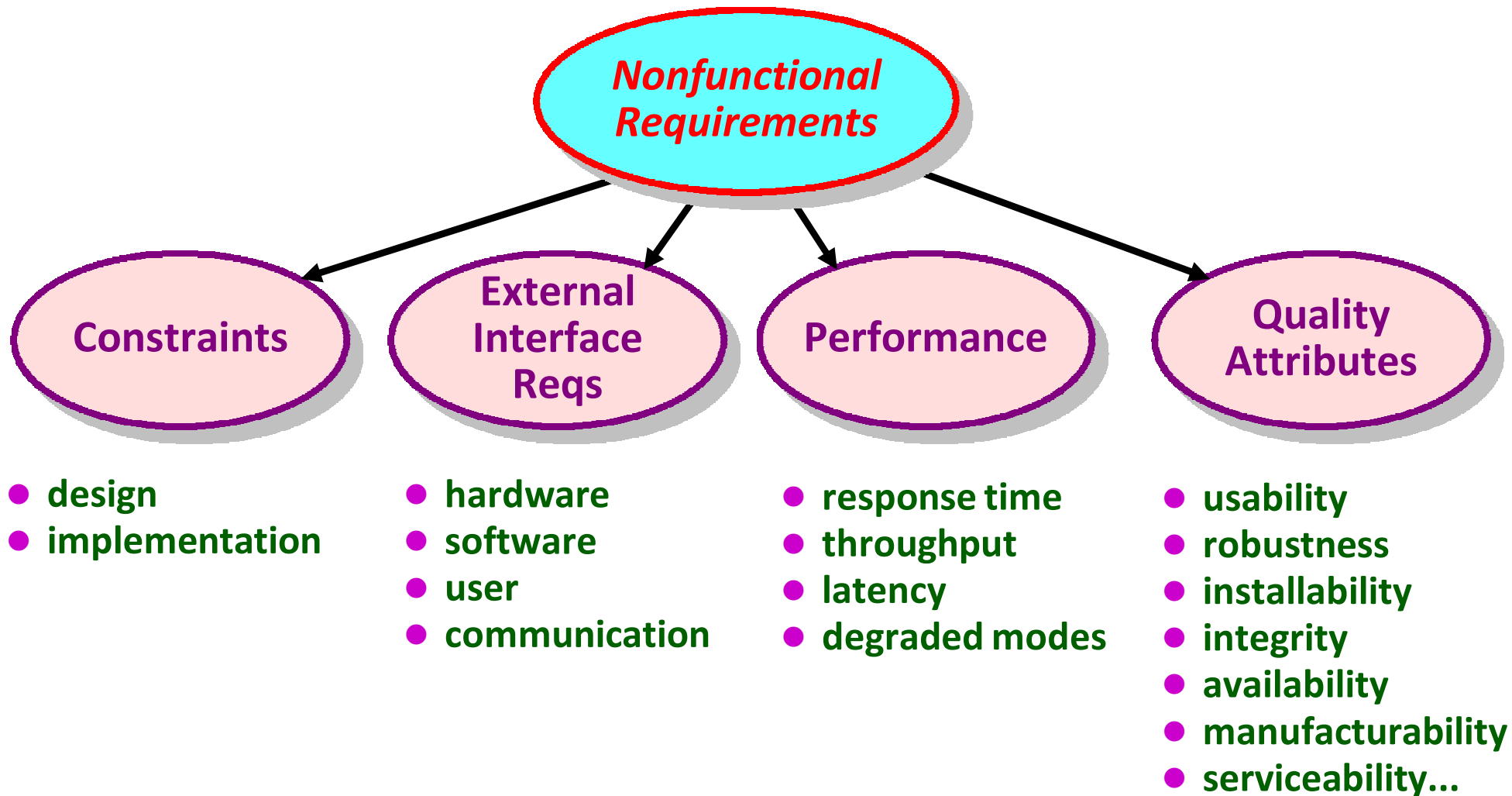
Firmware Application States

Application State	Application Control	Description
Off (0)		[functionality]
Power Up		[functionality]
Ready to Run	Go To Running	[functionality]
Running	- Go To ShutDown - Go to Standby	The firmware enters the 'Running' state on request of the host. This is obtained through the 'go to running' request from the host in the states 'Ready to Run', 'GoingStandby', 'Standby', 'GoingDown', 'Shutdown' and 'Error'. Immediately after the state is entered all data available in the firmware interface is sent to the host. If all data is send, the firmware notifies the host the state 'Running' is entered.
GoingStandby	Go To Running	[functionality]
Standby	Go To Running	[functionality]
GoingDown	Go To Running	[functionality]
Shutdown	Go To Running	[functionality]
Error	Go To Running	[functionality]

Sample State-Transition Diagram



Specify Nonfunctional Requirements



Implications of Quality Attributes

<i>Quality Attribute Types</i>	<i>Likely Technical Information Category</i>
Integrity, Interoperability, Robustness, Safety, Security, Usability	Functional requirements
Availability, Efficiency, Flexibility, Performance, Reliability	System architecture
Interoperability, Usability	Design constraint
Flexibility, Maintainability, Portability, Reliability, Reusability, Testability, Usability	Design guideline
Portability	Implementation constraint

Use Cases and Scenarios and Stories, Oh My!

Use Case: A description of a set of interaction sequences that a system performs to provide a result of observable or measurable value to one or more actors.

Scenario: One specific path through a use case; or, a story about a specific instance of a use case execution, perhaps with actual users identified and specific data.

User Story: “A user story describes functionality that will be valuable to either a user or purchaser of a system or software.” (Mike Cohn)

Abstraction in User Requirements

Postal Delight



**More
Abstract**

Prepare a mailing label

Prepare a mailing label to
send a package by second-
day UPS.

Chris wants to send a 2.5-pound package by
second-day UPS from Portland, OR, to Rochester,
NY. She wants it insured for \$75 and she wants a
return receipt. The package is marked fragile.

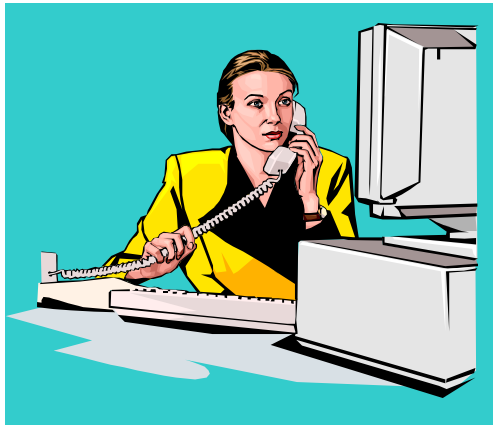
**Less
Abstract**

Use Cases and Functional Requirements

■ Two schools of thought:

- Use cases **are** the functional requirements.
- Use cases **reveal** the functional requirements.

User View

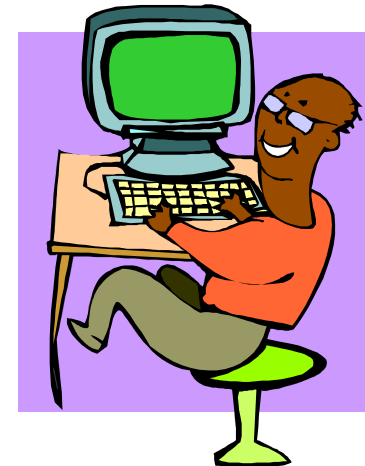


Use Cases

Analyst



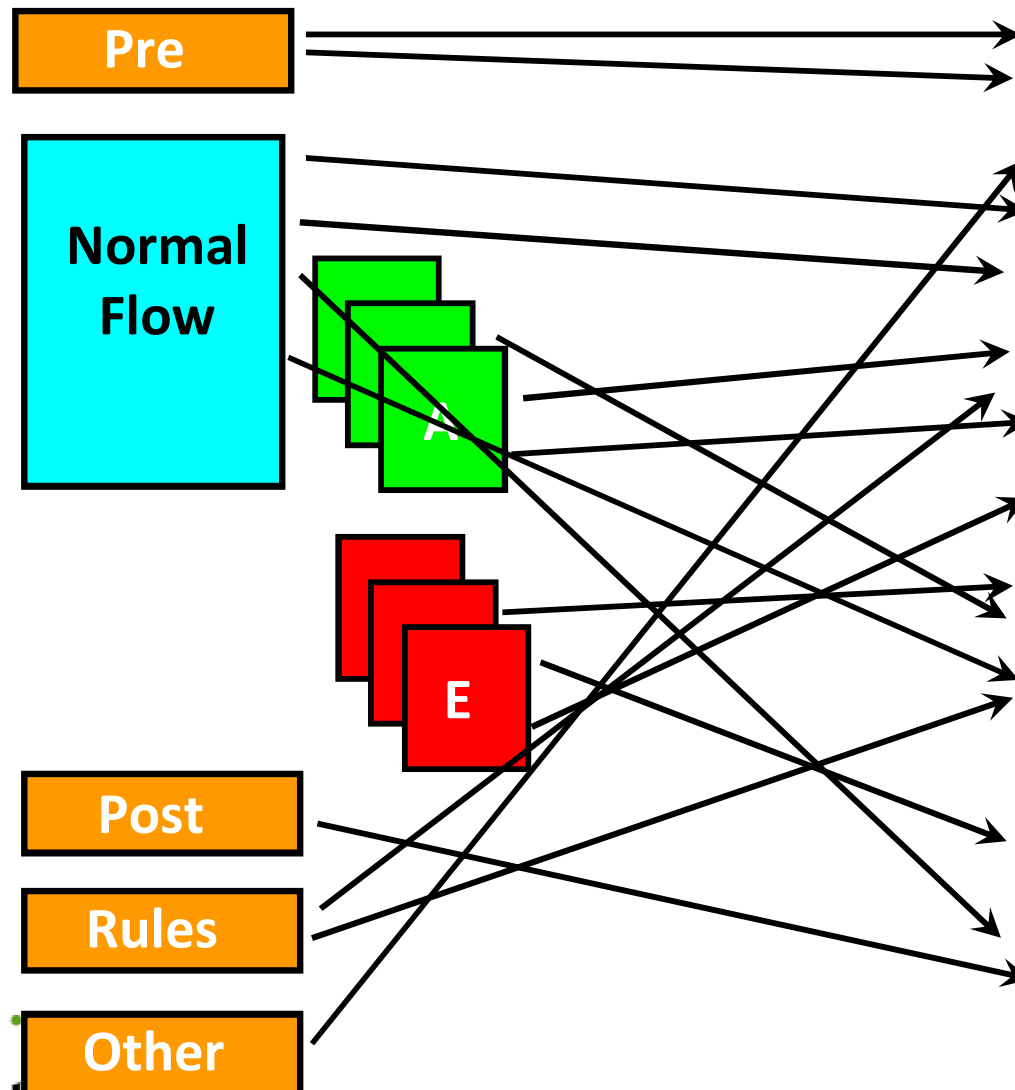
Developer View



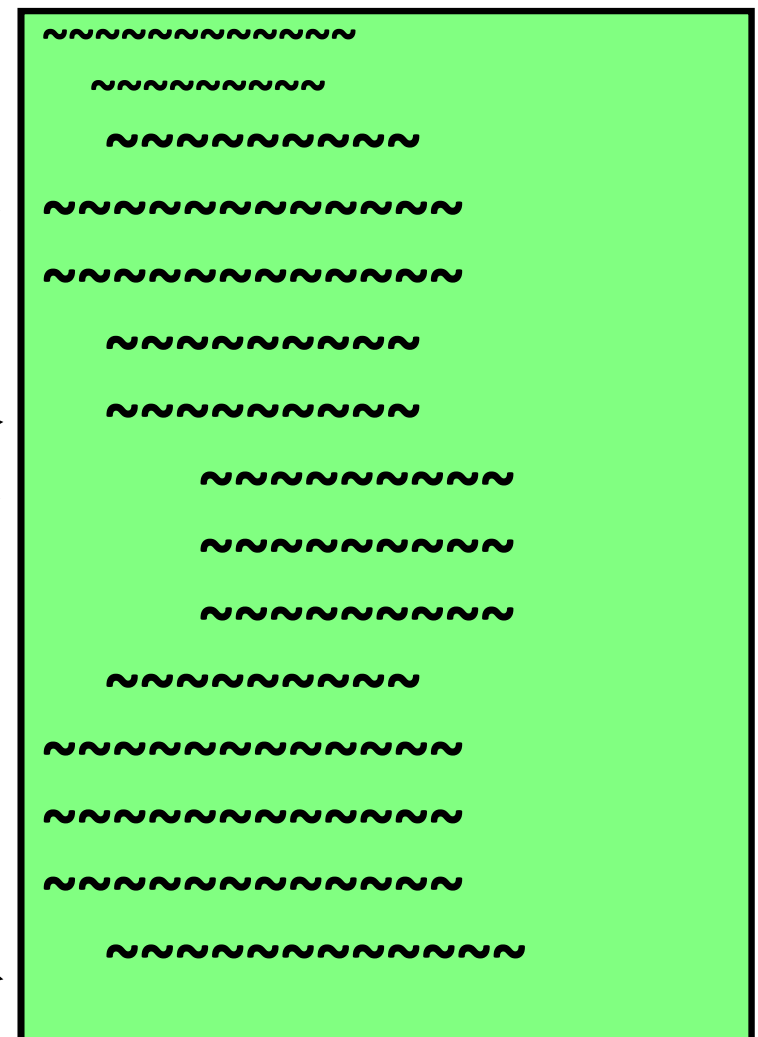
Functional Requirements

Organizing Information: Use Case and SRS

Use Case Organization



SRS Organization



Thought Process for Going From UCs -> FRs

■ Scan preconditions.

- what functionality is needed to test the precondition?
- what happens if a precondition isn't satisfied?

■ Scan dialog steps.

- what functionality will let the dialog take place?
- where are branches needed into alternative flows?

■ Scan exceptions.

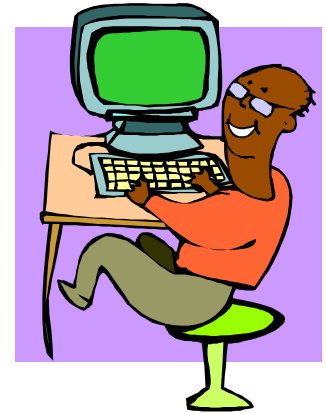
- write exception-handling reqs with the associated functional req
- what system recovery or reset operations might be needed?

■ Scan postconditions.

- how does the system satisfy visible and invisible postconditions?

■ Scan business rules.

- what functionality will enforce each business rule?



Thorny Issues in Software Requirements

***NO
SURPRISES!***



Sponsor: iRise

Register for the webinar on
1/30/13:

*How UPS and Others De-risked
Development with Visualization*

iRise.com/derisk

Download a 30 Day Free Trial

iRise.com/solutions

For more information call 800-
556-0399

- Simulate the full user experience quickly
- Validate the solution meets the business needs and is technically feasible – before code is written
- Generate consensus across teams in a collaborative environment
- Get a more precise estimation of resources, timeline, and overall project cost
- Generate functional specifications
- Generate UI code
- Increase user adoption, reduce change requests and avoid late cycle surprises.

