

Converting Use Cases into Technical Requirements – with Examples

Linda Westfall

lwestfall@westfallteam.com

www.softwareexcellenceacademy.com



Defining the Use Cases

Use cases identify how the system will be used

1. Define the system's scope
2. Identify the actors
3. For each actor, identify their interactions with the system
 - Main tasks or functions performed
 - Information acquired, produced, or changed
 - Information given to or wanted from the system
 - Notifications wanted from the system
4. Develop a use case for each interaction
5. Merge or create sub-use cases as appropriate

Use Case – Example

Use Case: Purchase Gas by Post-Paying Attendant

Actor Actions	System Responses
1. Customer selects to post pay attendant for gas	2. Prompts customer to select gas type
3. Customer selects gas type	4. Verifies sufficient inventory & sets gas pump to pump selected gas type & prompts customer to pump gas
5. Customer removes pump handle from holder & pumps gas	6. Monitors the gas flow, calculates & displays ongoing gallons pumped & price as gas is pumped
7. Customer returns pump handle to holder	8. Sends totals for gallons pumped & price to attendant & prompts customer to pay attendant

Use Case Alternatives – Example

Use Case: Purchase Gas by Post-Paying Attendant

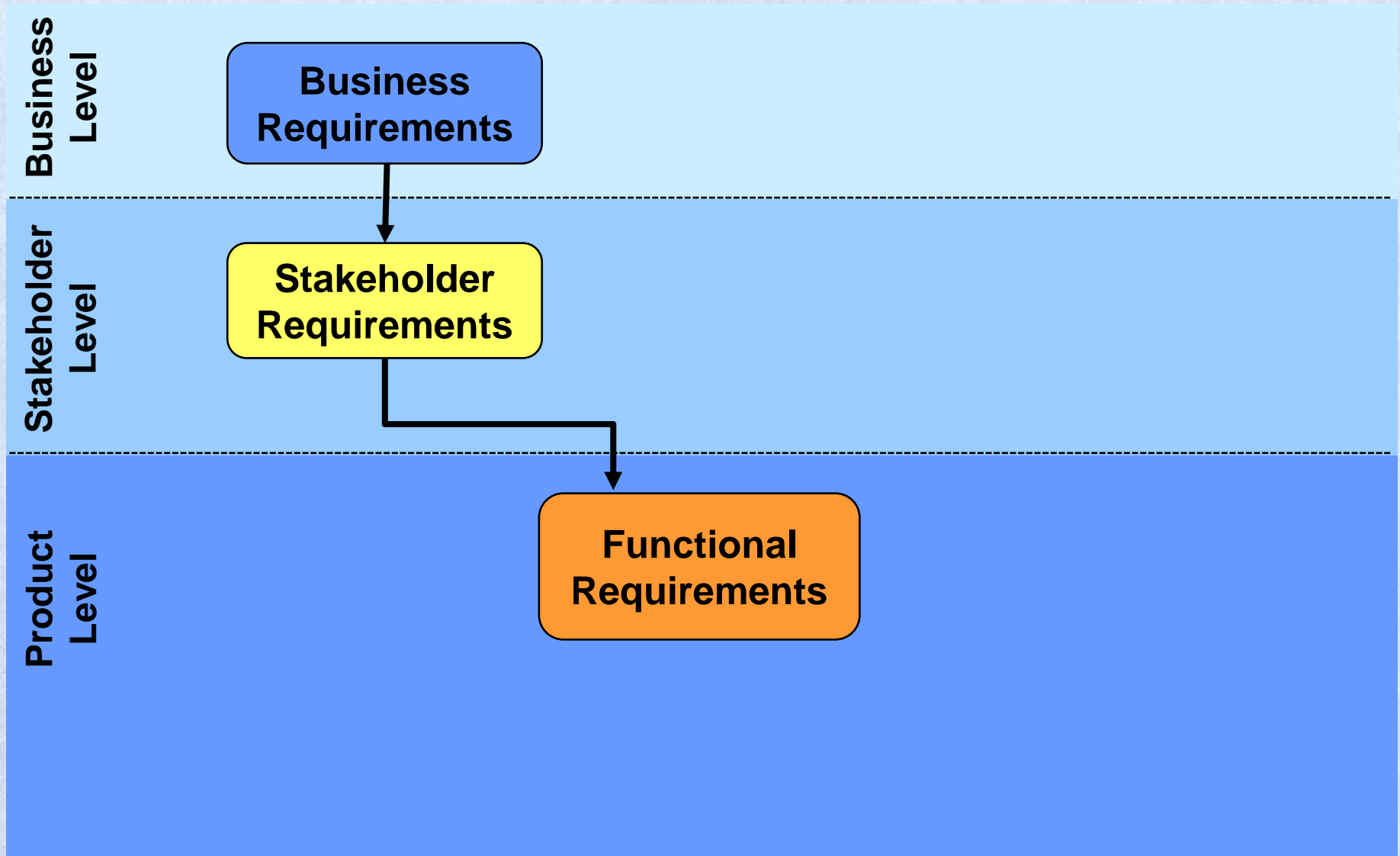
Actor Actions	System Responses
3a1. Customer selects to cancel transaction	3a2. Notifies attendant of cancellation, sends transaction cancel message to customer & returns pump to idle mode
5a1. Customer selects to cancel transaction	5a2. Notifies attendant of cancellation, sends transaction cancel message to customer & returns pump to idle mode

Use Case Exceptions – Example

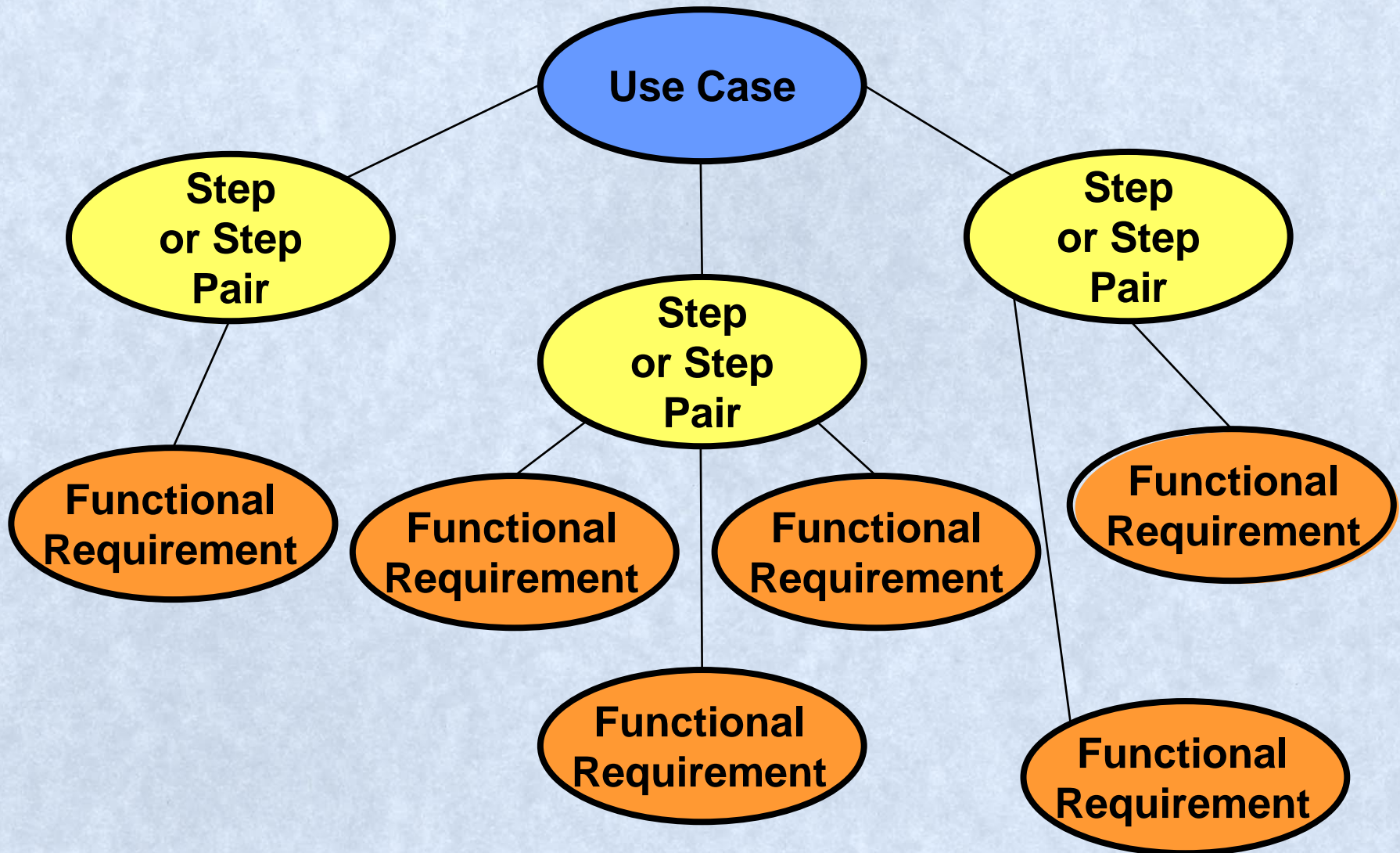
Use Case: Purchase Gas by Post-Paying Attendant

Actor Actions	System Responses
3b1. Customer responses with something other than a gas type or cancel	3b2. Notifies customer of invalid entry & re-prompts customer to select gas type 3b3. Return to step 3
Actor Actions	System Responses
	4a1. Identifies error indicating not enough gas of the type selected, reports error to customer & re-prompts customer to select gas type 4a2. Return to step 3

Levels & Types of Requirements



Use Case → Functional Requirements

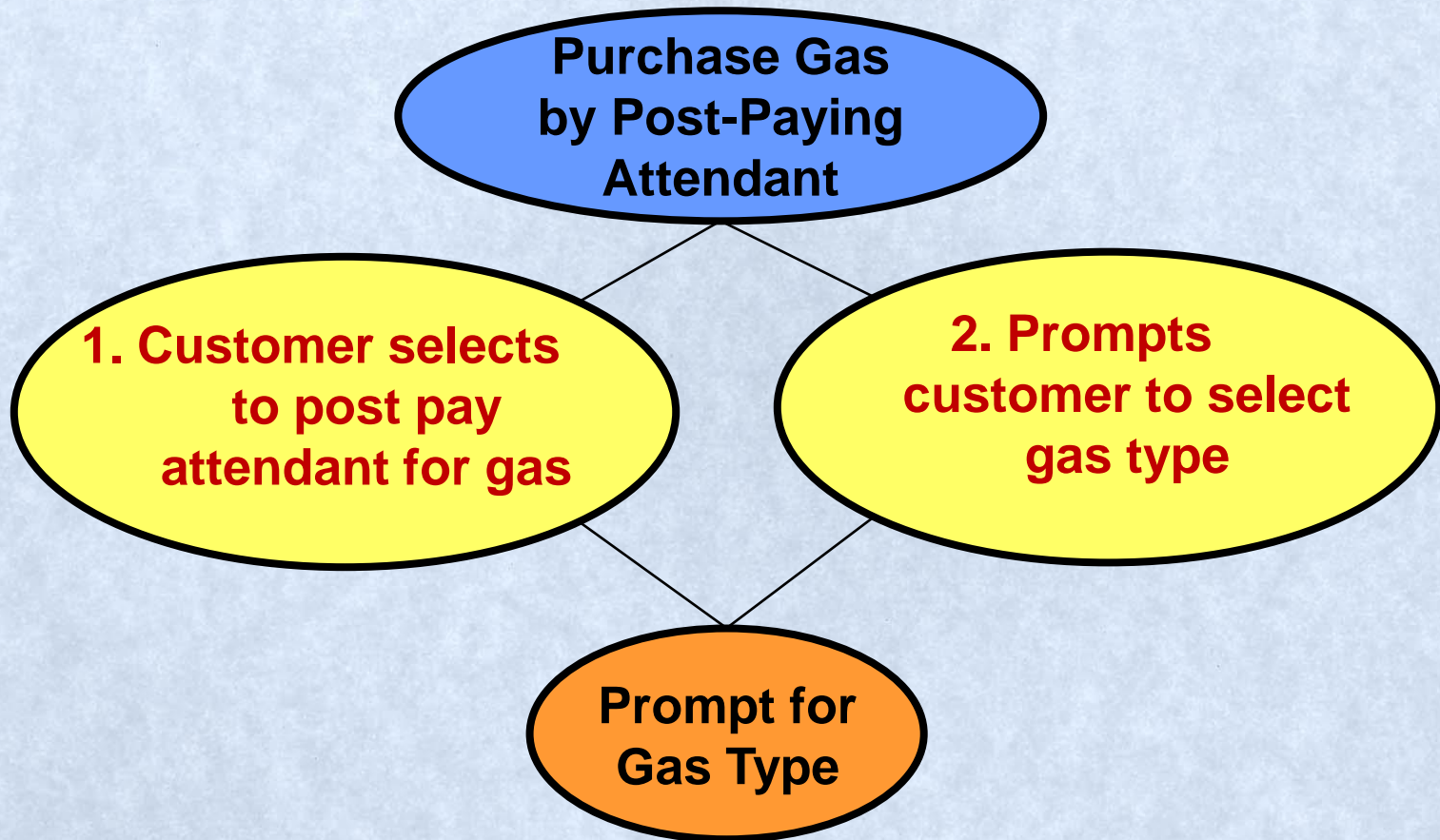


Step Pair – Example #1

Use Case: Purchase Gas by Post-Paying Attendant

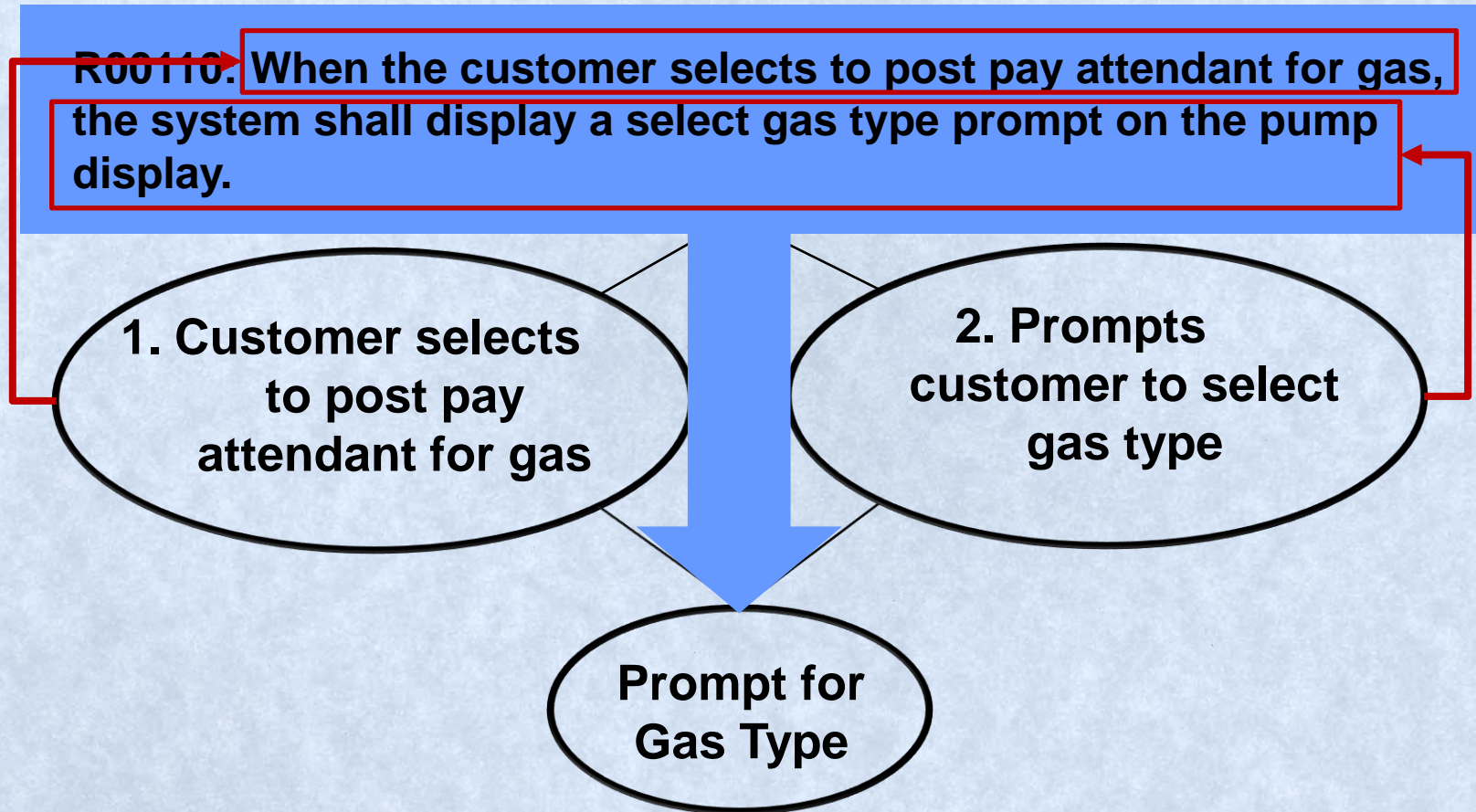
Actor Actions	System Responses
1. Customer selects to post pay attendant for gas	2. Prompts customer to select gas type
3. Customer selects gas type	4. Verifies sufficient inventory & sets gas pump to pump selected gas type & prompts customer to pump gas
5. Customer removes pump handle from holder & pumps gas	6. Monitors the gas flow, calculates & displays ongoing gallons pumped & price as gas is pumped
7. Customer returns pump handle to holder	8. Sends totals for gallons pumped & price to attendant & prompts customer to pay attendant

Step Pair – Example #1



Functional Requirements Template: [When] [condition/event], the [subject/noun] shall [do what].

Step Pair – Example #1



Functional Requirements Template: [When] [condition/event], the [subject/noun] shall [do what].

Step Pair – Example #2

Use Case: Purchase Gas by Post-Paying Attendant

R00115: When gas is being pumped, the system shall poll the gas flow control hardware once every 10th of a second to determine the cumulative *Amount_Of_Gas_Pumped* on that pump.

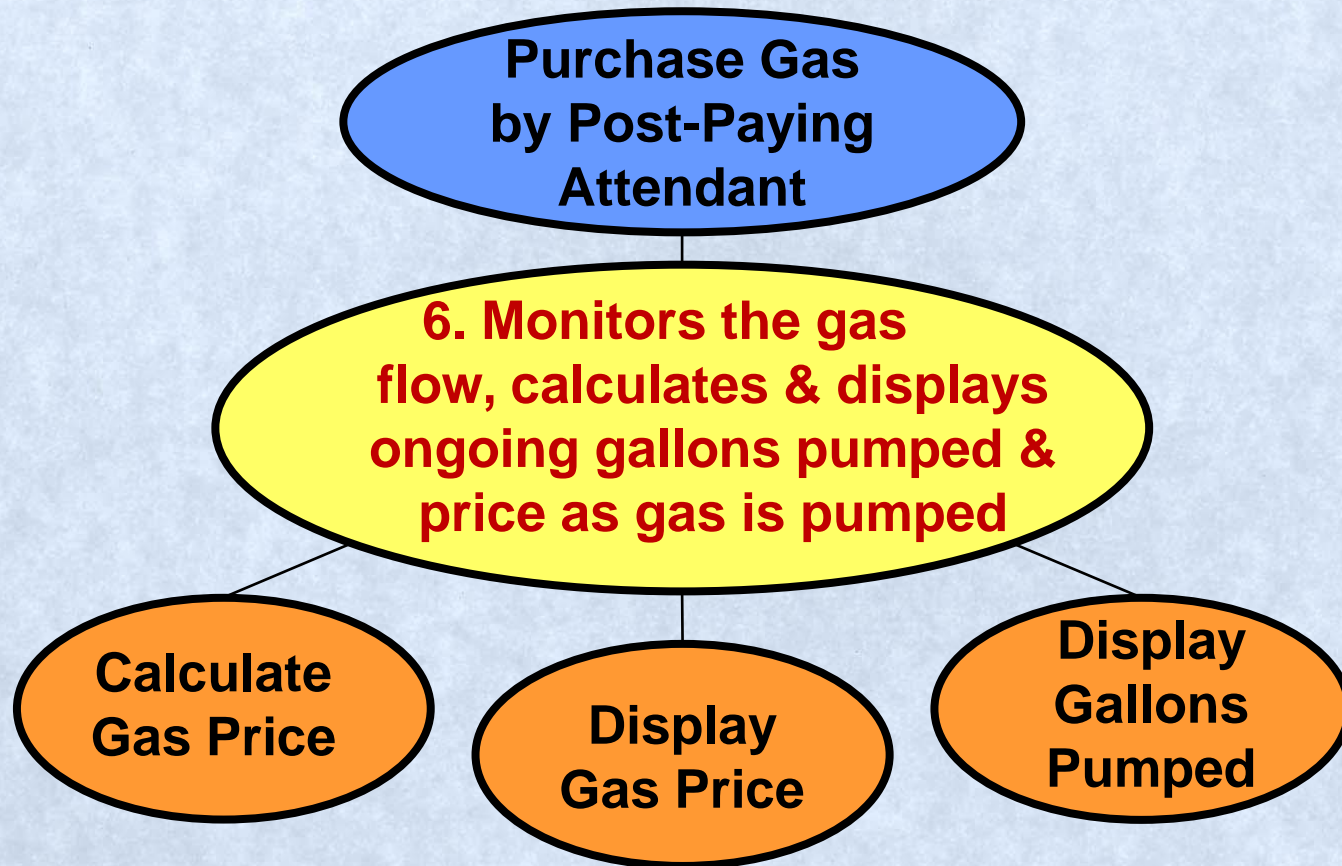
- | | |
|--|---|
| 1. Customer inserts gas attendant handle | 2. Attendant prompts for gas type |
| 3. Customer selects gas type | 4. Verifies sufficient inventory & sets gas pump to pump selected gas type & prompts customer to pump gas |
| 5. Customer removes pump handle from holder & pumps gas | 6. Monitors the gas flow, calculates & displays ongoing gallons pumped & price as gas is pumped |
| 7. Customer returns pump handle to holder | 8. Sends totals for gallons pumped & price to attendant & prompts customer to pay attendant |

Single Step – Example #3

Use Case: Purchase Gas by Post-Paying Attendant

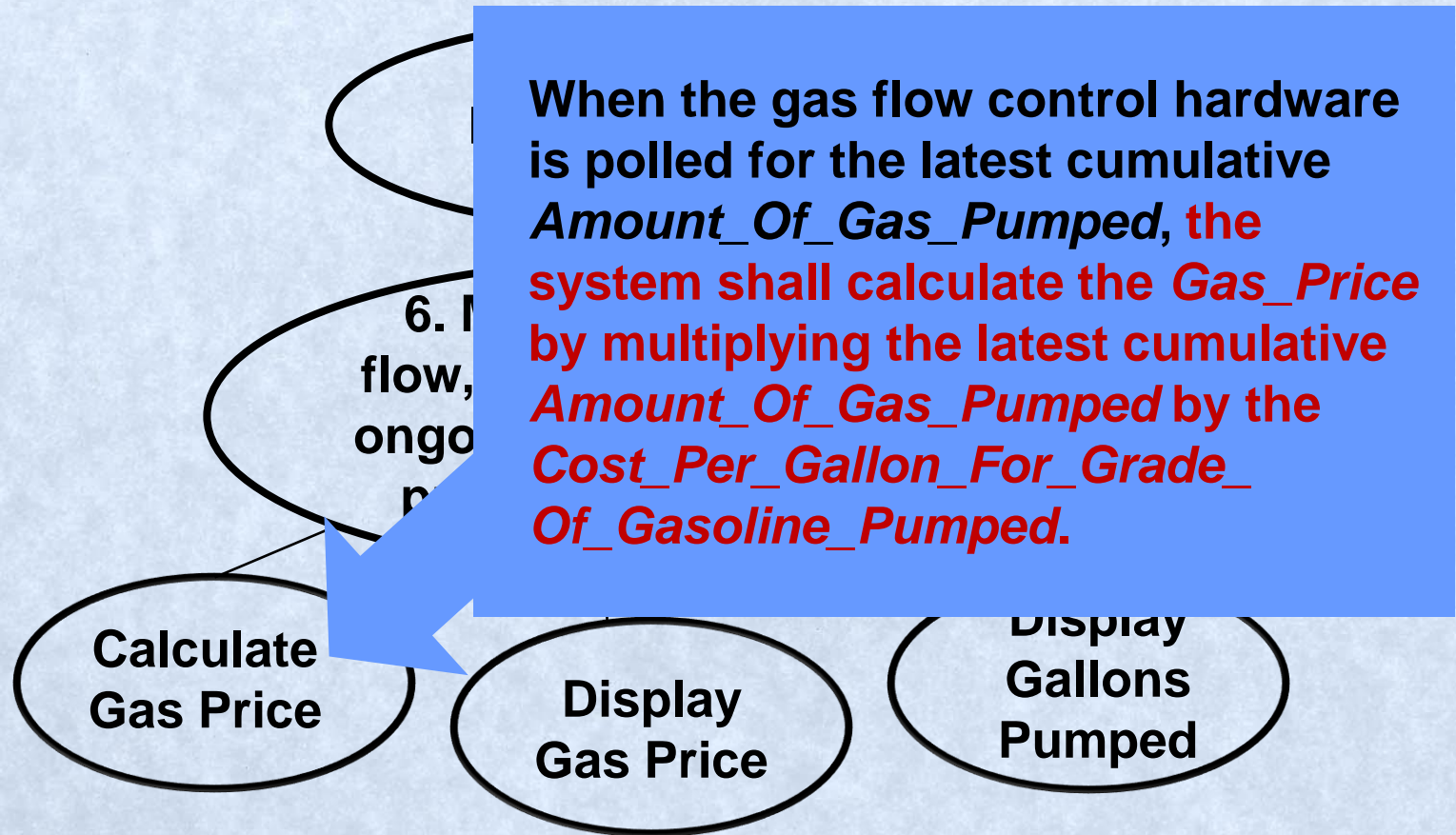
Actor Actions	System Responses
1. Customer selects to post pay attendant for gas	2. Prompts customer to select gas type
3. Customer selects gas type	4. Verifies sufficient inventory & sets gas pump to pump selected gas type & prompts customer to pump gas
5. Customer removes pump handle from holder & pumps gas	6. Monitors the gas flow, calculates & displays ongoing gallons pumped & price as gas is pumped
7. Customer returns pump handle to holder	8. Sends totals for gallons pumped & price to attendant & prompts customer to pay attendant

Single Step – Example #3



Functional Requirements Template: [When] [condition/event], the [subject/noun] shall [do what].

Single Step – Example #3

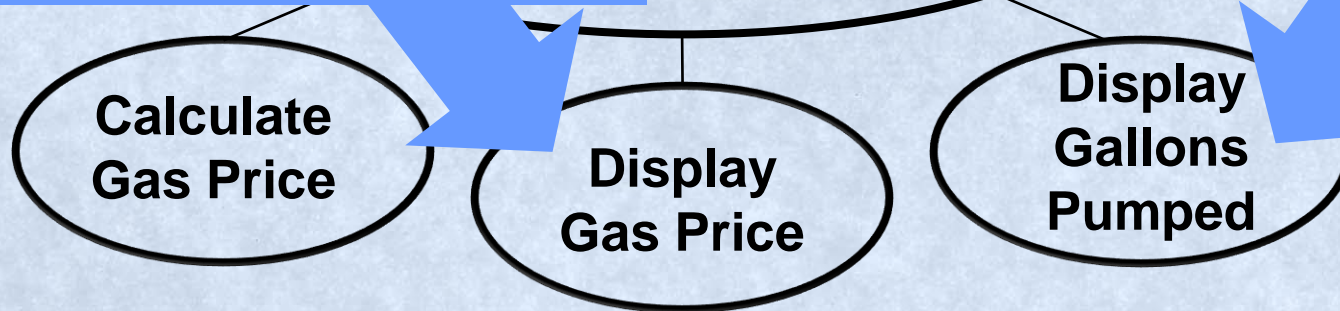


Functional Requirements Template: [When] [condition/event], the [subject/noun] shall [do what].

Single Step – Example #3

When the gas flow control hardware for a pump is polled for the latest cumulative *Amount_Of_Gas_Pumped*, the system shall display the most recent calculated *Gas_Price* in US dollars on that pump's display.

When the gas flow control hardware for a pump is polled for the latest cumulative *Amount_Of_Gas_Pumped*, the system shall display the latest cumulative *Amount_Of_Gas_Pumped* on that pump's display.



Functional Requirements Template: [When] [condition/event], the [subject/noun] shall [do what].

Another Functional Requirements Template

Rnnn [When] [condition/event], the [subject/noun] shall:

- Rnnn.1 [do what]
- Rnnn.2 then [do what]
- Rnnn.3 then [do what]
- ...

Single Step – Example #3

R00116: When the gas flow control hardware is polled for the latest cumulative amount of gas pumped, the system shall:

R00116.1: Calculate the *Gas_Price* by multiplying the latest cumulative *Amount_Of_Gas_Pumped* by the *Cost_Per_Gallon_For_Grade_Of_Gasoline_Pumped*.

R00116.2: Then display the most recent calculated *Gas_Price* in US dollars on that pump's display.

R00116.3: Then display the latest cumulative *Amount_Of_Gas_Pumped* on that pump's display.

5. Customer removes pump handle from holder & pumps gas
7. Customer returns pump handle to holder

6. **Monitors the gas flow, calculates & displays ongoing gallons pumped & price as gas is pumped**
8. Sends totals for gallons pumped & price to attendant & prompts customer to pay attendant

Other Functional Requirements Templates

Rnnn [When] [condition/event], if [status, event or condition] the [subject/noun] shall [do what].

Rnnn [When] [condition/event], if [status, event or condition] the [subject/noun] shall:

- Rnnn.1 [do what]
- Rnnn.2 then [do what]
- Rnnn.3 then [do what]
- ...

Single Step – Example #4

Use

3b1

R00112: When the system is verifying sufficient gas inventory, if the system identifies an error indicating not enough gas of the selected type to pump, the system shall display an error message indicating that there is not enough gas of the selected type to pump and a select gas type re-prompt on the pump's display.

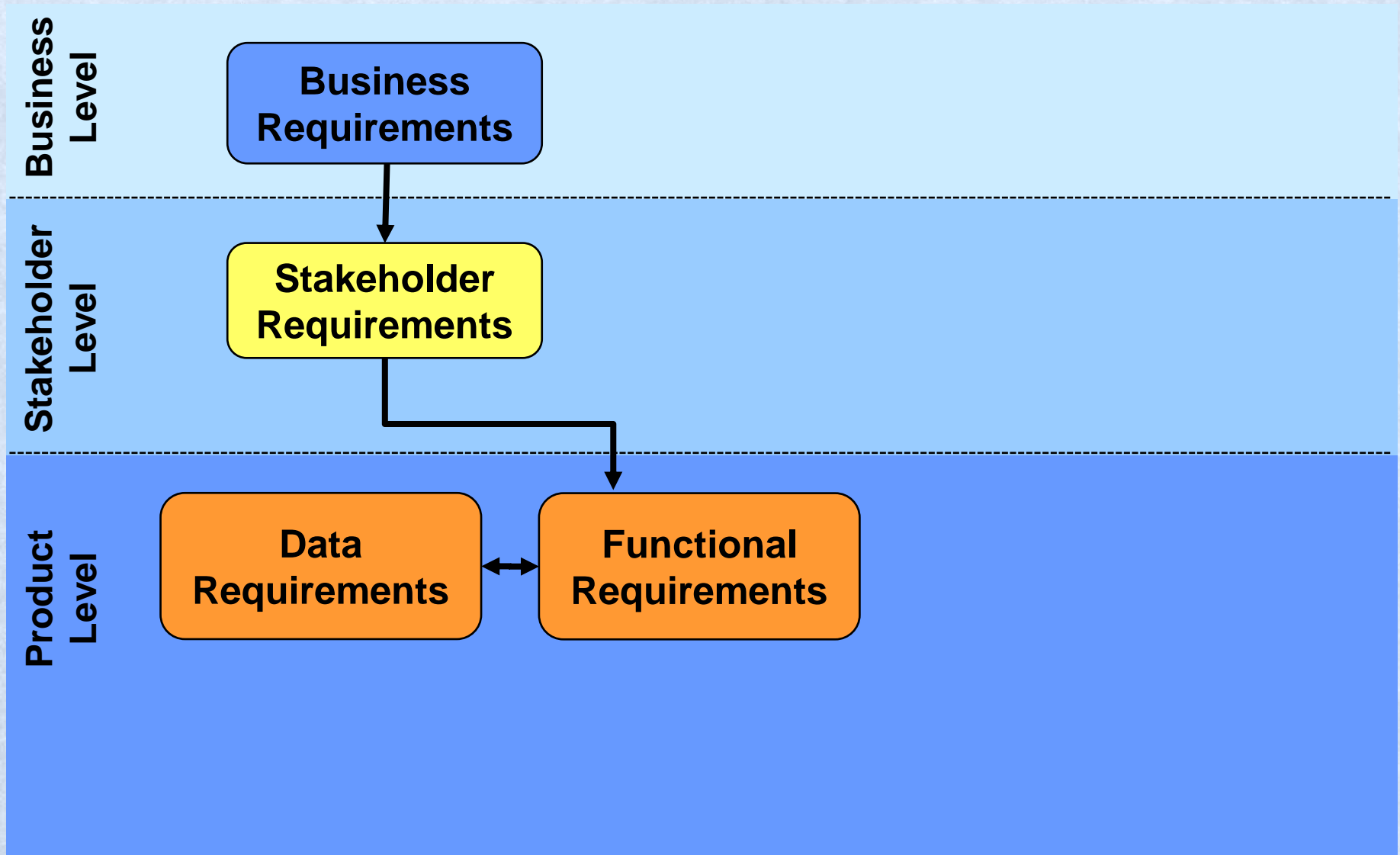
Actor Actions

System Responses

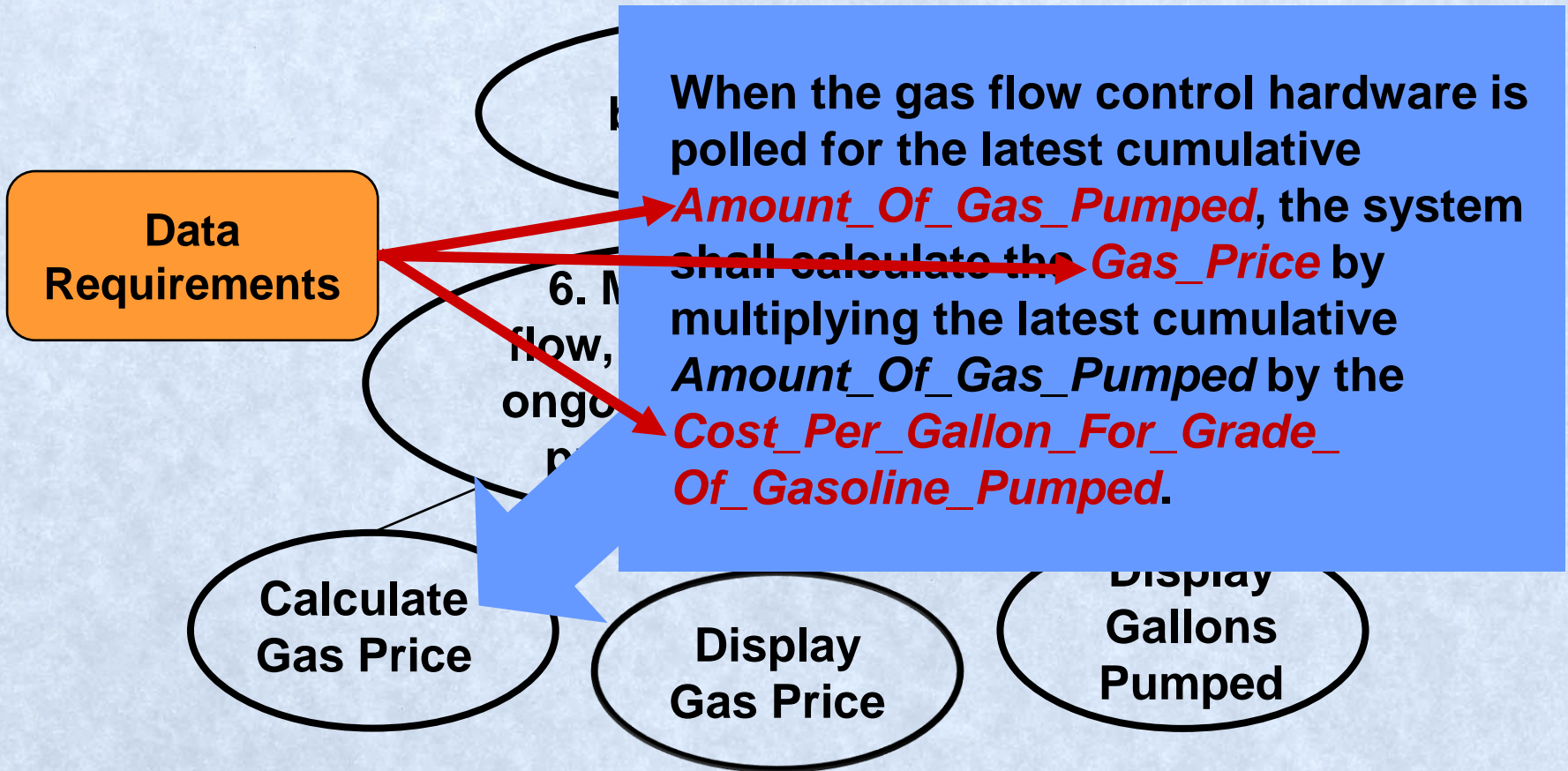
4a1. Identifies error indicating not enough gas of the type selected, reports error to customer & re-prompts customer to select gas type

4a2. Return to step 3

Levels & Types of Requirements



Data Requirements – Example #3



Data Requirements – CRUDL

For each data element identified, consider whether functional requirements are needed for:

Creating the data element

Reading the data element

Updating the data element

Deleting the data element

Listing the data element

Data Element Requirements – Examples

Cost_Per_Gallon_For_Grade_Of_Gasoline_Pumped. :

Creating the data element

Reading the data element

Uppd R00234: When a new Gas_Type is created, the system shall:

Del R00234.1 Prompt the user to enter the
Cost_Per_Gallon_For_Grade_Of_Gasoline_Pumped

List R00234.2 Then accept the user's inputted value

R00234.3 Then verify the inputted *value*

R00234.4 Then, if the inputted value is invalid, the system shall report an error and return to requirement

R00234.1

R00234.5 Else if the value is valid, the system sets the
Cost_Per_Gallon_For_Grade_Of_Gasoline_Pumped
for the new gas type to the inputted value.

Data Requirements – CRUDL

Cost_Per_Gallon_For_Grade_Of_Gasoline_Pumped. :


Creating the data element

Reading the data element

Updating the data element

Delet

Listin



R00235: When the manager polls the system for the *Cost_Per_Gallon_For_Grade_Of_Gasoline_Pumped*, the system shall display a list of current gas types and their *Cost_Per_Gallon_For_Grade_Of_Gasoline_Pumped* for each of those gas types on the manager's display.

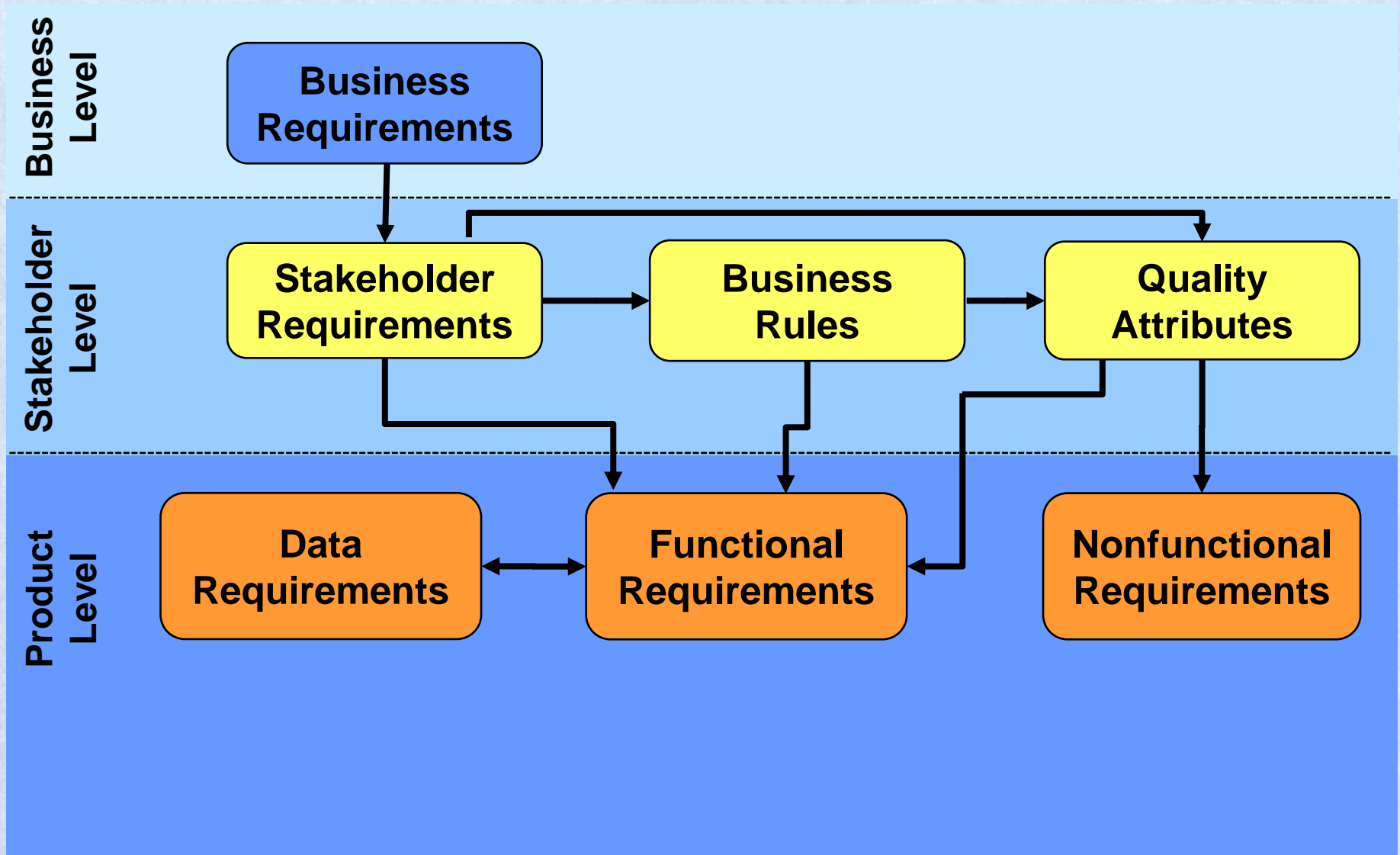
R00236: When a pump is placed reset, the system shall refresh the pump display with the current *Cost_Per_Gallon_For_Grade_Of_Gasoline_Pumped* for each gas type available from that pump.

Data Management Requirements

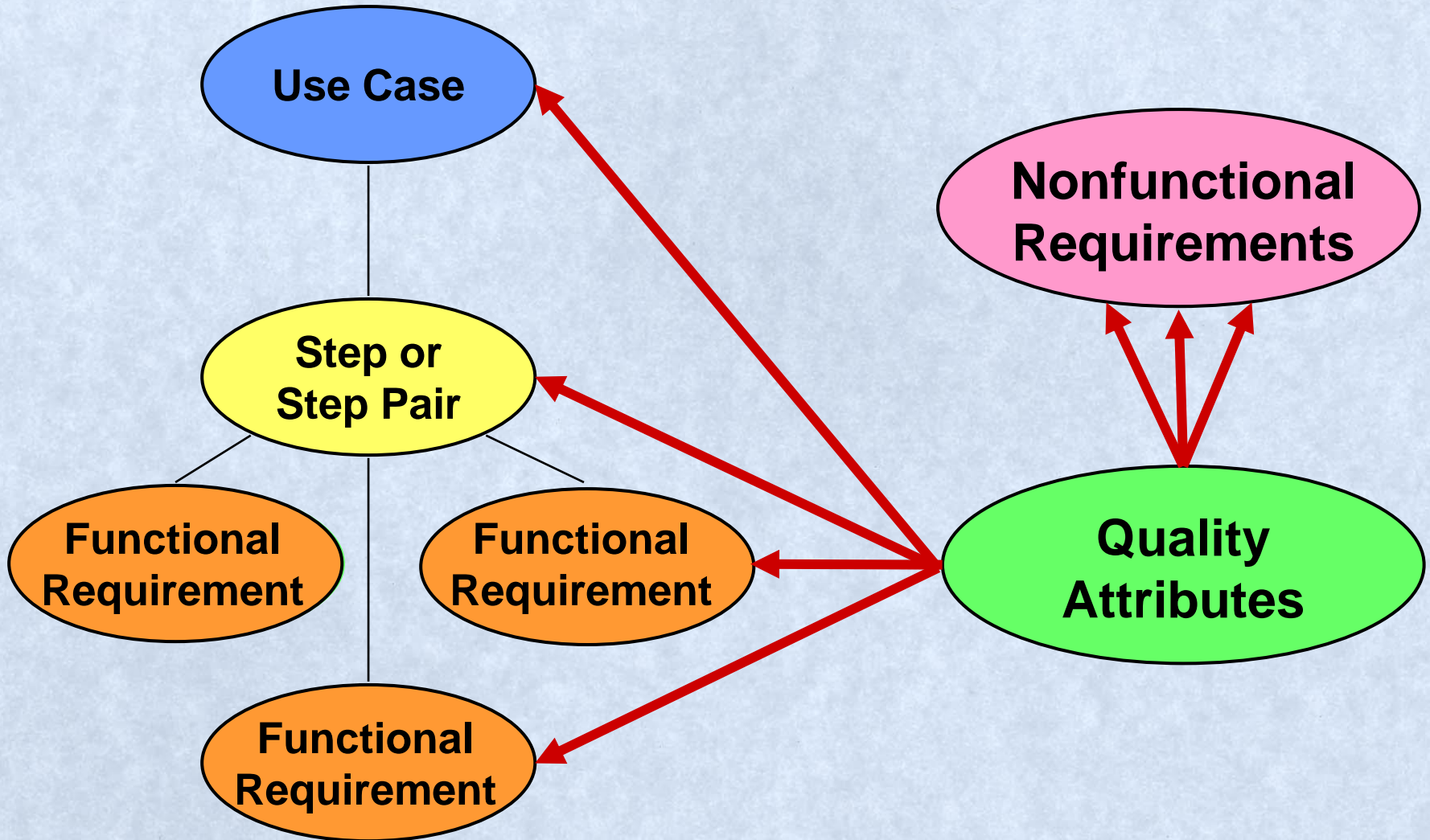
Examples of other data management requirements include:

- Data access: Specify data access methods for direct real-time access, batch access or data as a service
- Data refreshing: Specify how data from clients or other remote devices will be refreshed/updated into the centralized data store
- Data sharing: Specify requirements for sharing data with other systems or for data interoperability
- Performance: Specify any performance requirements associated with the data
- Metadata: Specify any required metadata (data about the data), including audit trails of data item access

Levels & Types of Requirements



Use Case → Nonfunctional Requirements



Quality Attributes

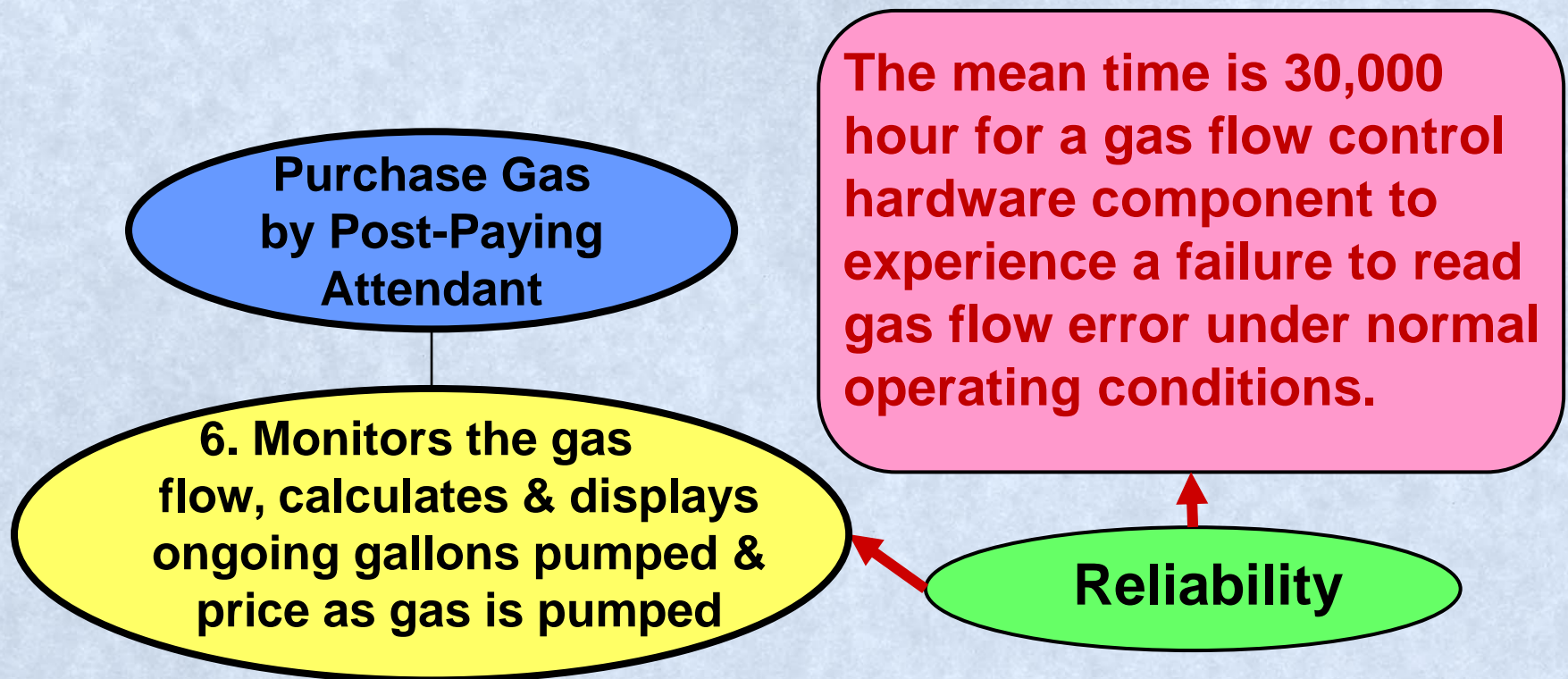
Quality attributes are the characteristics that define the software product's quality, for example:

- Reliability
- Availability
- Performance
- Usability
- Security (Integrity)
- Safety
- Efficiency
- Interoperability
- Accuracy
- Accessibility
- Installability
- Flexibility
- Robustness
- Maintainability
- Reusability
- Testability
- Portability
- Supportability

Reliability Requirements

Reliability nonfunctional requirement template:

[mean time] for a [defined product or component] to experience [defined failure type] under [defined conditions]. [based on Gilb-05]



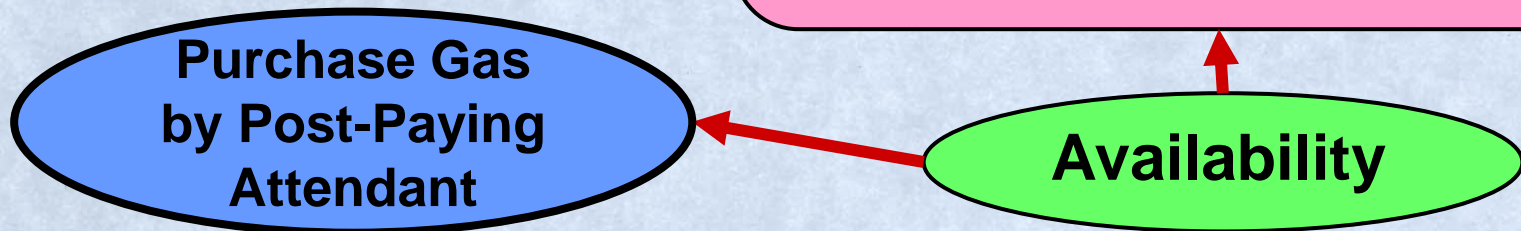
Availability Requirements

Availability nonfunctional requirement templates:

[percentage] of defined [time period] a defined [product] is [available] for its defined [tasks].

A [defined product or product component] will experience [an average or a maximum amount of time] of [defined failure type] during a [time period] under [defined conditions].

A pump will experience no more than an average of 3 minutes of unplanned outage during a 12

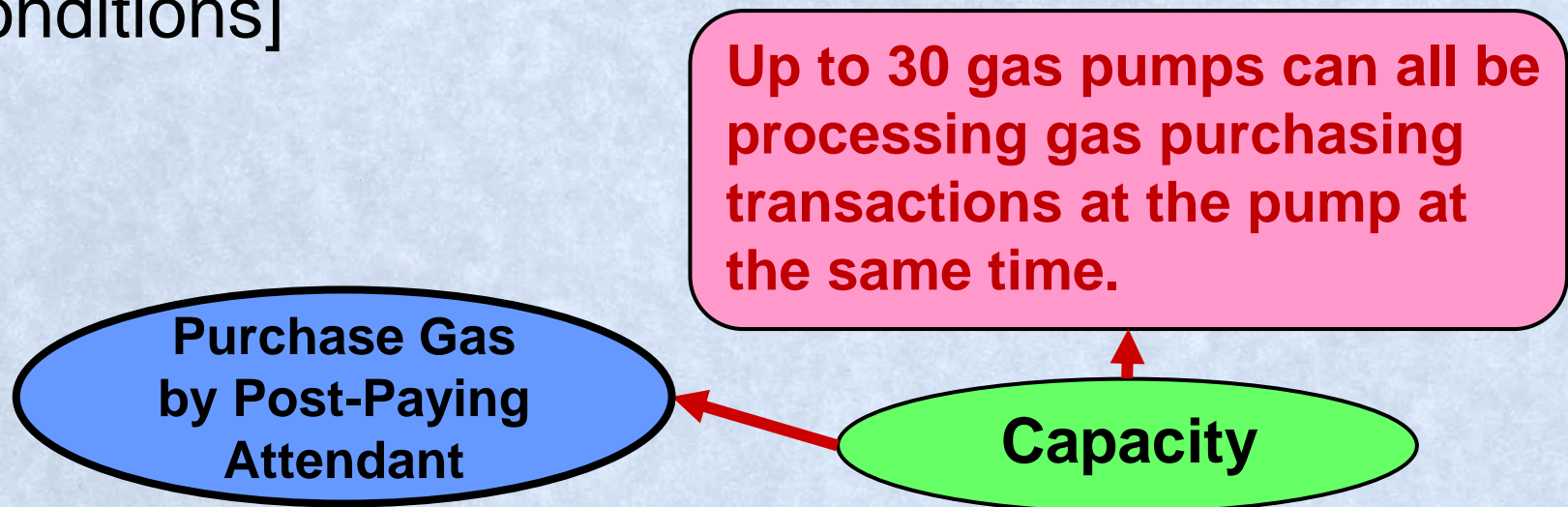


Performance Requirements

Performance nonfunctional requirement templates:

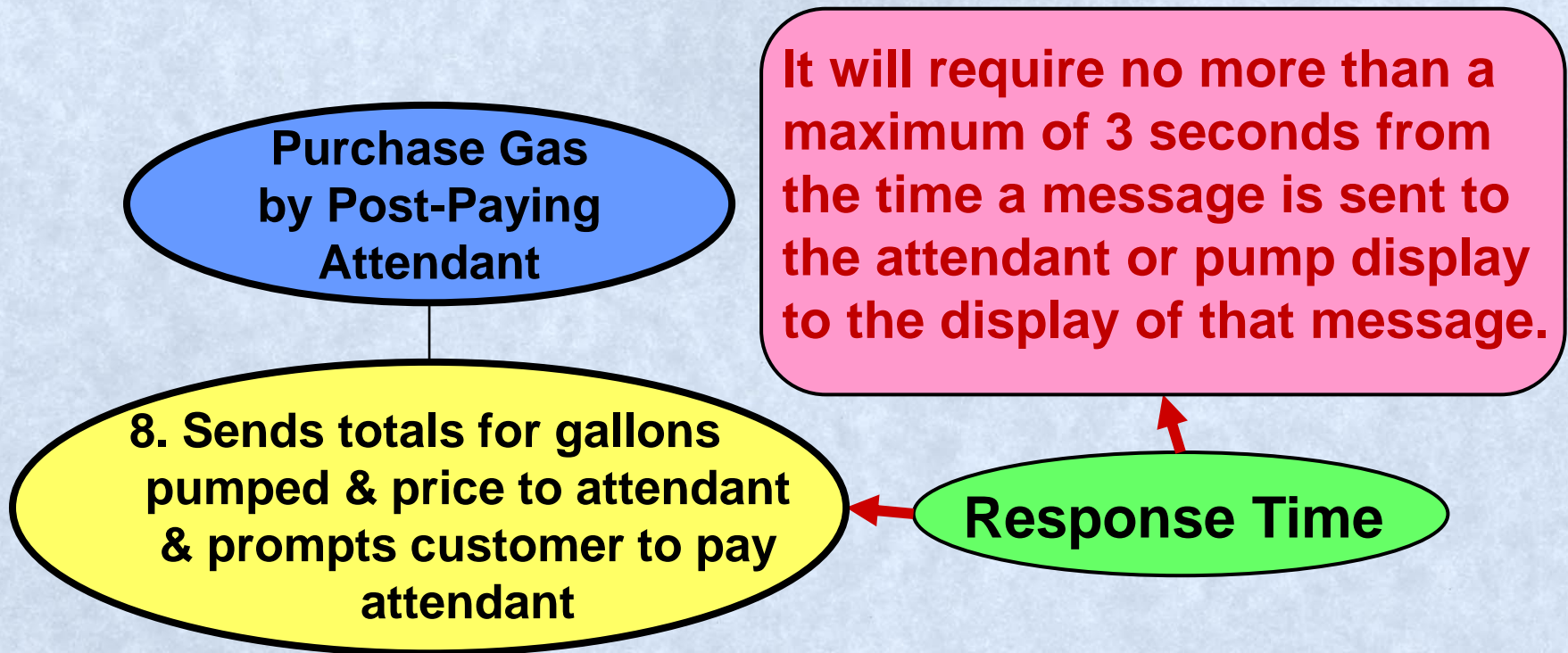
Throughput: [quantity] of [defined work units] which can be successfully handled per [time unit]

Capacity: [quantity] of [defined entities or components] can be [task, action or state] under defined [conditions]



Performance Requirements

Response time: [mean time or maximum time] of a defined [response] to [a defined event]



Usability Requirements

Usability nonfunctional requirements templates:

Ease of learning: [effort or time] required to become proficient in performing a [task or action] under defined [conditions]

Productivity: [quantity] of defined [tasks or actions] that can be performed using the product per [time unit] under defined [conditions]

Response time: [mean time or maximum time] of a defined [response] to a defined [user stimulus]

Error rate: [quantity] of user [error type] mistakes experienced per [time unit] under defined [conditions]

Likeability: [quantity or percentage] of users that report liking the product

Security Requirements

[probability] that a defined [product or product component] shall [handle (e.g., detect, prevent, recover from)] a defined [attack] under defined [conditions]

The systems shall [lock, place out of service] a [external access] after [number] unsuccessful attempts to [login, handshake, authenticate] within [time period]

The system shall log all attempts to [access secure data, perform specific function] by users having insufficient privileges

The system shall [time out, log out, lock] access to [data, asset, function] after [time period] of inactivity

Safety Requirements

The system shall [prevent, detect, report, react to] a [type of failure] such that [type of accident] is prevented under [defined conditions]

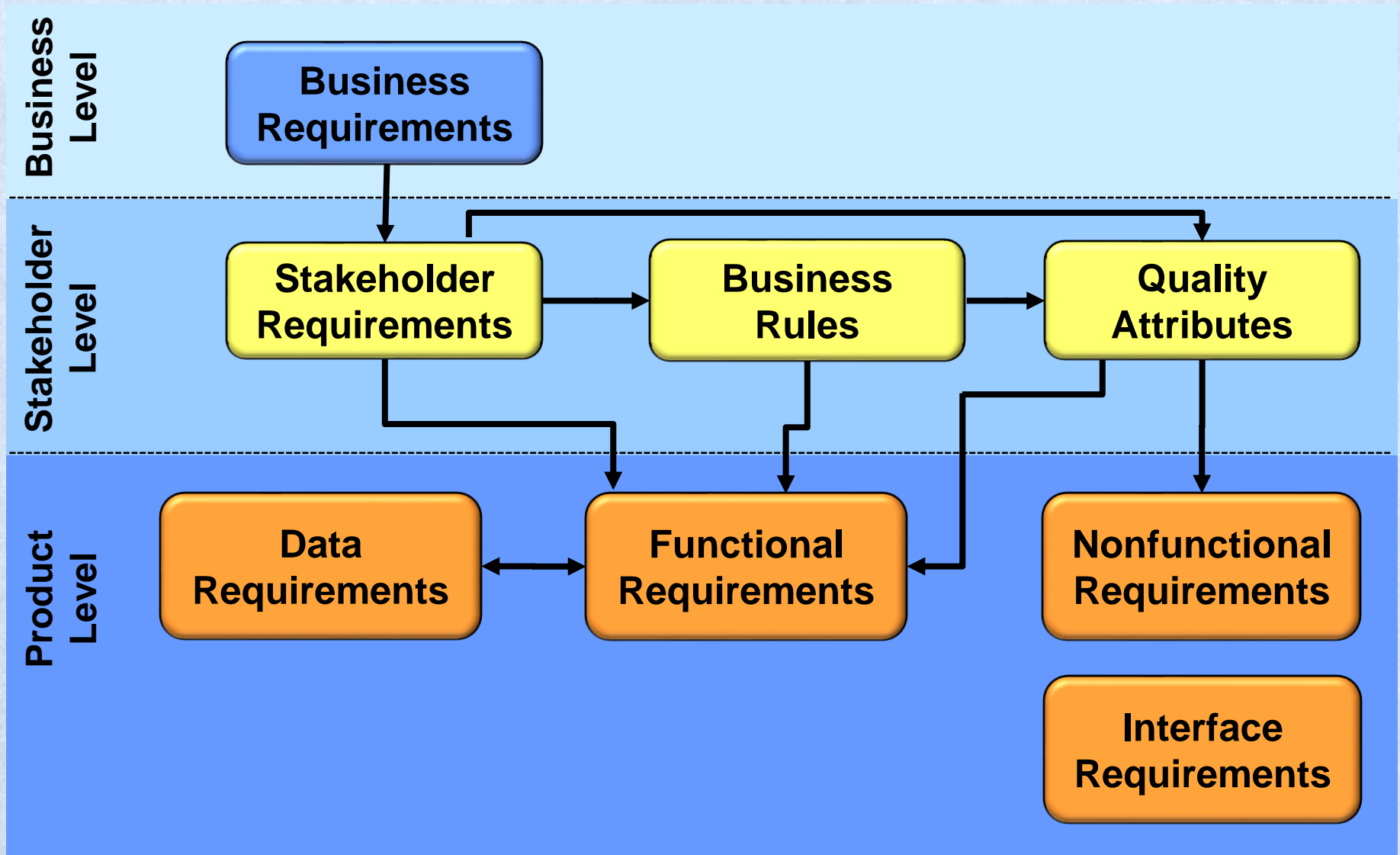
The system shall not cause more than [average or maximum number] of [type of safety incidents] per [time interval or number of events] under [defined conditions]

The system shall not allow [a safety risk] to exceed [specified level or variance] under [defined conditions]

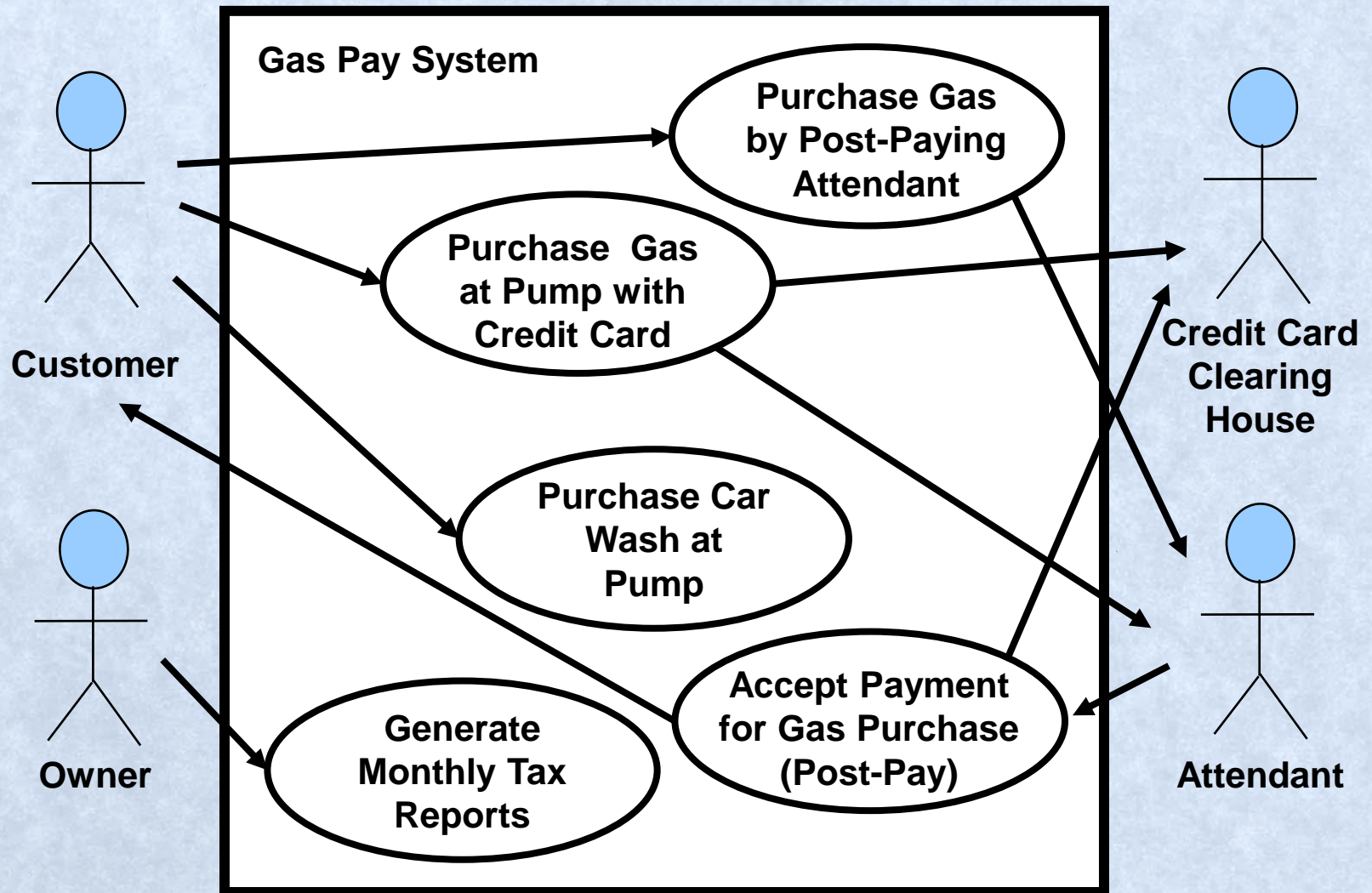
The system shall react to [type of accident] by performing [type of action]

The system shall [prevent, detect, react to] [type of accident] [percentage of the time]

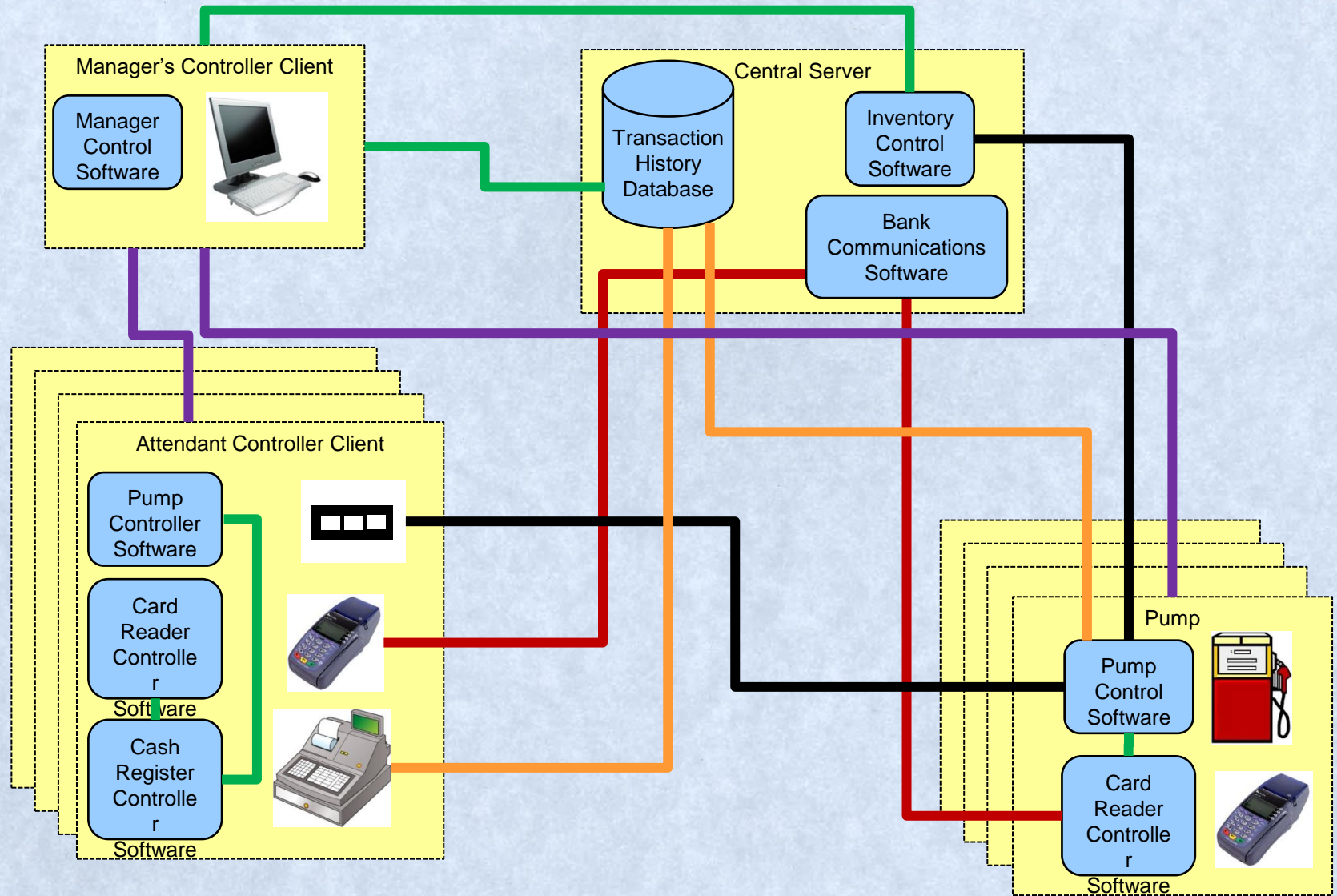
Levels & Types of Requirements



External Interface Requirements



Internal Interface Requirements



Interface Requirements – Examples

Types of information specified for interface requirements include:

- Data or messaging types, formats, or structures, including valid values or ranges
- Control values
- Interrupts, communication priorities or interactions
- Data translations or bit mappings
- The use of standardized communications protocols
- User interface style guides for usability or product branding purposes

Interface Requirements – Examples

- Other nonfunctional requirements affecting the interfaces (for example, transfer rates, response times, frequencies, capacity, security controls, handshaking)
- Reports, prompts & the reporting of errors
- Devices supported

Presenter: Linda Westfall

More than 45 years in software:

- President of the Westfall Team
- Sr. Manager of Quality Metrics & Analysis, Manager of Production Software, software process engineer, software engineer & systems analyst

Active professionally:

- ASQ Software Division past chair, ASQ Certification Board, PMBOK® contributor & P.E. exam development
- P.E. Software Engineering, ASQ Fellow, CSQE, CMQ/OE & CQA, PMP, Certified Scrum Master, Lean Six-Sigma Black Belt, ISTQB Certified Tester
- Author: *The Certified Software Quality Engineer Handbook, 2nd Edition*



References

- [Gilb-05] Tom Gilb, *Competitive Engineering: A handbook for Systems Engineering, Requirements Engineering , and Software Engineering Using Planguage*, Elsevier Butterworth Heinemann, Amsterdam, 2005.
- [Wiegers-03] Karl E. Wiegers, *Software Requirements, 2nd Edition*, Microsoft Press, Redmond, WA, 2003.
- [Westfall-17] Linda Westfall, *The Certified Software Quality Engineer Handbook, 2nd Edition*, ASQ Quality Press, Milwaukee, WI, 2017.

Contact Information



Linda Westfall

phone: (972) 867-1172

email: lwestfall@westfallteam.com

www.softwareexcellenceacademy.com