

How to write good requirements

Module 4 of 10

Converting stakeholder wants to needs

Session 1 of 2

Version 1.3.6

How to write good requirements

0401-1



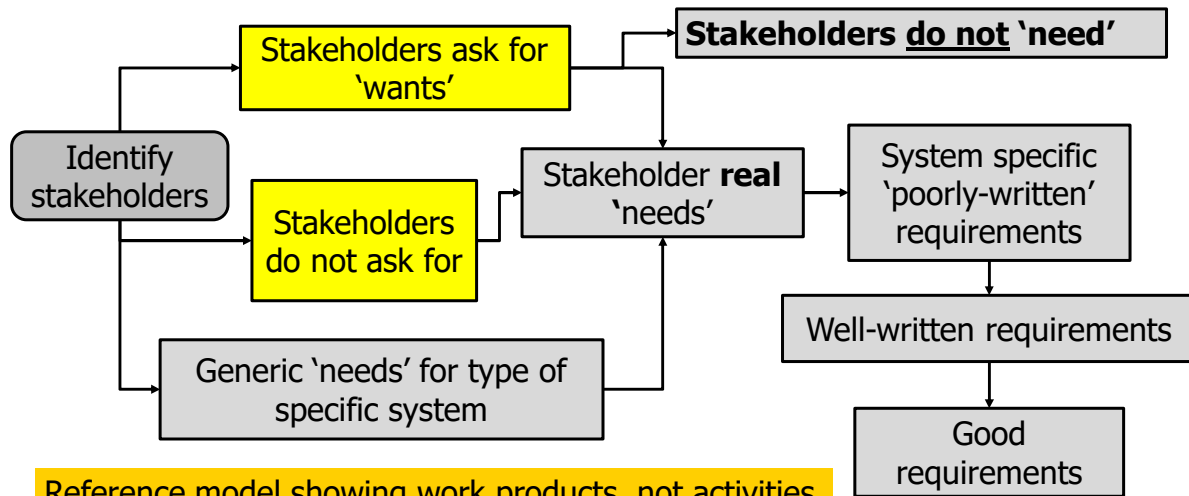
Course Module topics

1. Introduction to requirements
2. Stakeholders and their importance
3. Communicating with the stakeholders
- 4. Converting stakeholder wants to needs**
5. Documenting stakeholders' needs
6. Converting stakeholder needs to requirements
7. Converting requirements to well-written requirements
8. Converting well-written requirements to good requirements
9. The use of requirements in the rest of the system development process
10. Summary and closeout

How to write good requirements

0401-2

Gap analysis for writing good requirements



The top 5 reasons why you can't write a good requirement

0301-3

Objectives of Module 4

1. To explain how to convert stakeholder functional and performance "wants" to "needs"
2. To explain the difference between functions and misuse functions
3. To introduce risk management
4. To explain need for prioritization of needs and how to prioritize them
5. To explain three ways to maximize the completeness of the needs
6. To explain how to influence the stakeholders to want the system they need
7. To explain how to determine if the need is for Commercial-off-the-Shelf (COTS) equipment
8. To provide the opportunity to exercise the 5 levels of knowledge in the updated Blooms taxonomy

How to write good requirements

0401-4

Knowledge components

- Lecture
 - Sets the context and provides overview
- Readings
 - 0402 Systems Thinker's Toolbox Section 13.1: Checkland's Soft System Methodology
 - ~~0403 Kasser, J.E., Applying Holistic Thinking to the Problem of Determining the Future Availability of Technology, IEEE Transactions on Systems, Man, and Cybernetics: Systems, Volume 46, Number 3, 2016.~~
- Exercises
 - 4-11 Scenarios to requirements
 - 4-21 Knowledge reading 0402
 - 4-22 Knowledge reading 0403 (optional)

How to write good requirements

0401-5

Module topics

- **Converting wants to needs**
- An introduction to Misuse Functions (risks and risk management)
- Prioritization of needs
- Converting needs to functions
- Scoping for affordable cost and realistic schedules
- Ways of influencing stakeholders
- When the need is for COTS
- Exercises



How to write good requirements

0401-6

Converting "wants" to "needs"

- "Need" = "Want" (if it is a real need)
 - Validate the reason for the need
- Generic needs shall be adapted to specific situation
- Customer may have to be educated before approving adapted generic "need"
 - Reason for the generic "need"
 - Consequences of not incorporating it in system
- Needs the stakeholders have not asked for (wanted)
 - Treat as tailored generic "need" (educate)
- Stakeholder insists on a "want" that is not a "need"
 - Don't argue, just document it as a "need" if stakeholder does not agree to withdraw request
 - Explain the reason and consequences to the customer
 - Let the customer argue with the stakeholder, or set it to a low priority

How to write good requirements

0401-7

Determining the real need-1

- Create generic CONOPS (to-be) of the future (reference representation)
 - Without undesirability of current situation
 - With improvements stakeholders want
- Use domain knowledge from similar systems (past and present)
- Include generic needs
- Discuss generic needs with the customer, other stakeholders may not be aware of them, nor might they need to become aware
- List assumptions
- Validate correctness in assumptions
- Don't assume you know better than the stakeholders
- Update generic CONOPS to become the specific CONOPS

How to write good requirements

0401-8

Determining the real need-2

- Validate each stakeholder requirement request (want) against specific CONOPS
- Don't show the specific CONOPS to the stakeholders, show sketches the stakeholder can understand
- If the request does not show up as a function in the specific CONOPS discuss the reason for the request with the originating stakeholder
 - If the need is valid modify the specific CONOPS to meet the request.
 - If the need is not valid, then discuss don't argue
 - If stakeholder does not agree to withdraw request, document it and flag it for discussion with the customer

How to write good requirements

0401-9

Impact assessment

- Prioritise the requirement (request)
- Determine if a contradiction exists
- Estimate cost to implement
- Determine cost drivers
- Perform sensitivity analysis
- Negotiate requirement with customers (and other stakeholders) based on sensitivity analysis
 - Are cost drivers really necessary?
- Document decisions in requirement repository

Creating Outstanding Systems Engineers

0702-10



Acceptance criteria

- How shall we know when the requirement has been fulfilled?
 - Identifies both acceptance criteria and real need
 - Helps to write both requirement statement and acceptance criteria
 - Can be used to clarify existing poorly written (ambiguous) requirements when planning tests

Systemic and systematic requirements engineering

0401-11



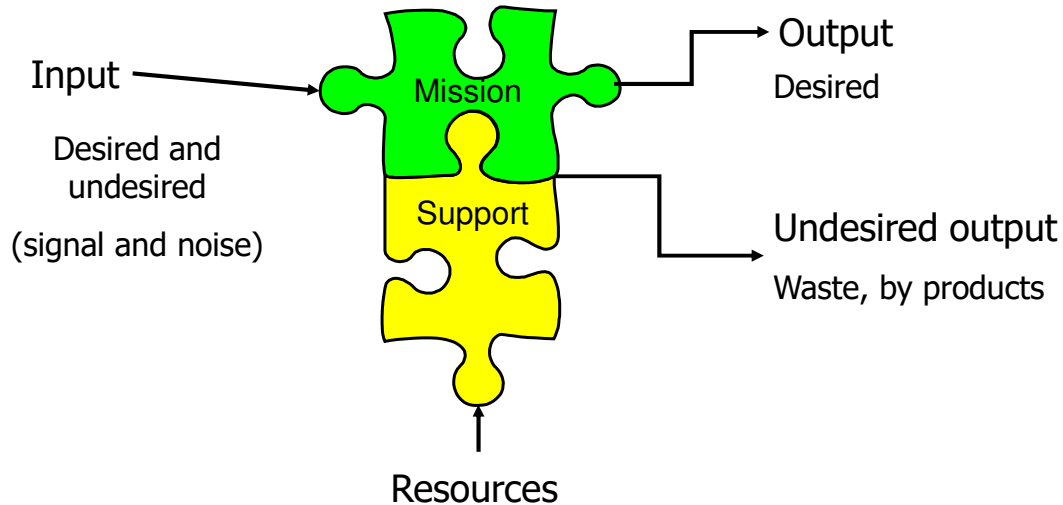
The three functional representations

1. Each **representation** contains quantified functional scenarios including 'don't cares'
 1. Current situation (as-is)
 2. CONOPS (to-be)
 3. Transition process
2. Compare wants/needs to generic CONOPS and transition representations
 - Checklands' SSM, see Reading 0402, can be tailored to situation
3. Adjust CONOPS and transition representations to include specific needs, creating the specific CONOPS

How to write good requirements

0401-12

System functional template (stakeholders)



How to write good requirements

0401-13

Generic support functions include ...

- Stakeholders may not want or know they need
 - Backing up and recovering data
 - Complying with pertinent laws and regulations
 - Managing changes in needs
 - Maintaining and repairing
 - Monitoring and reporting
 - Ensuring safety
 - Ensuring security
 - Training
 - Disposing of waste (safely and in compliance to regulations)

How to write good requirements

0401-14

Three ways to maximize completeness

1. Ensure the generic inputs and outputs of each scenario actually interface to another scenario (if appropriate)
 - Input (desired and undesired), output (desired and undesired), control and report
 - E.g., perhaps input and store functions exist, output function does not
 - N^2 chart is a useful tool
2. Ensure the generic processes are performed in each scenario
 - Storing, processing
3. Use templates
 1. Useful for related scenarios and functions
 2. E.g. if there is a "collecting statistics" scenario, there should also be scenarios for storing, retrieving and reporting on the statistics

How to write good requirements

0401-15

Module topics

- Converting wants to needs
- **An introduction to Misuse Functions (risks and risk management)**
- Prioritization of needs
- Converting needs to functions
- Scoping for affordable cost and realistic schedules
- Ways of influencing stakeholders
- When the need is for COTS
- Exercises



How to write good requirements

0401-16

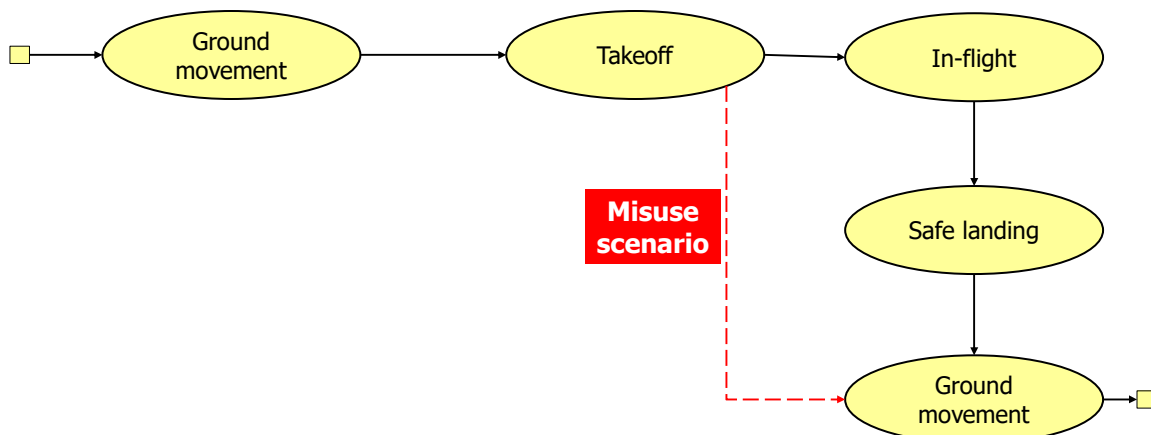
Misuse functions in each scenario

- Functions are assumed to be performed to meet the need
- Something could go wrong (misuse function or risk) in each scenario
- Active Brainstorm what could go wrong in each scenario (risk identification)
- Assign unique reference number to each risk
 - Internally (failure)
 - As a result of an undesired external input
- Assign probability of occurrence and severity of impact if it occurs
- How and where (in which scenario) to mitigate or prevent the risk
- If prevented move to inactive status (show its been identified and prevented)
- Update the appropriate scenarios or add new ones with the mitigation or prevention functions
 - Scenario rational includes "preventing Risk TBS from occurring in scenario TBS")
- Builds (functional) risk management into process up-front

How to write good requirements

0401-17

Scenario mapping (Flight)

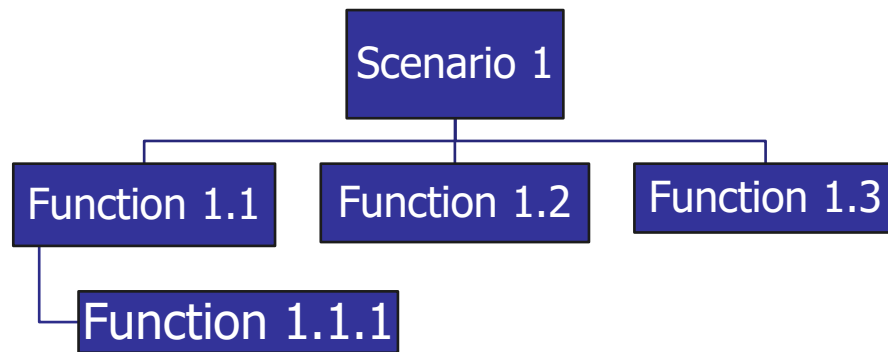


How to write good requirements

0401-18

Converting scenarios to functions

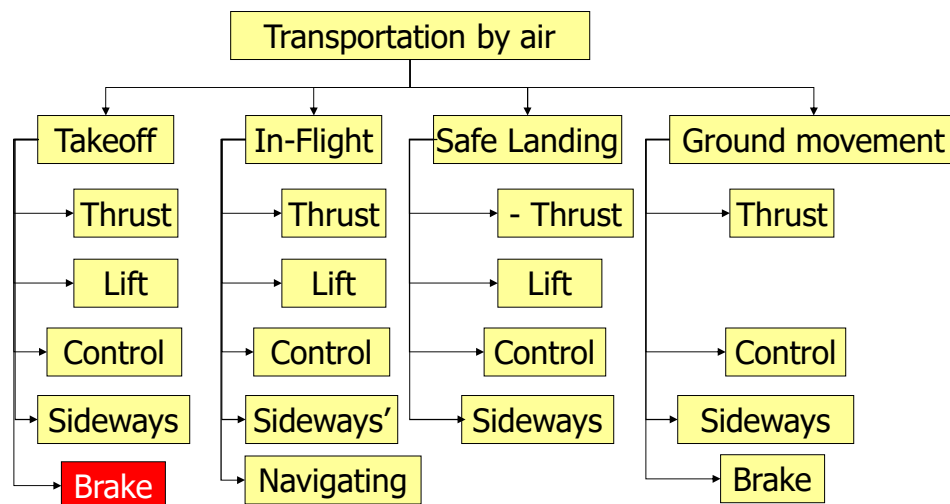
- Use Principle of Hierarchies



How to write good requirements

0401-19

Example: Scenarios/functions of flight



How to write good requirements

0301-20

Scenario to function mapping

	Takeoff	In-flight	Safe landing	Ground movement
Thrusting	X	X	(reverse) X	X
Lifting	X	X	X	
Braking	X (misuse)			X
Controlling	X	X	X	X
Moving sideways	X	X	X	
Navigating		X		
Communicating	X	X	X	X

- Functions can be reused in different scenarios
- Table tool can be used at design time for function to component mapping

How to write good requirements

0401-21

Scenario to function mapping (cup of coffee)

- Scenario making a cup of coffee
- Functions in reverse order
 - Stirring the mixture in the cup
 - Adding the ingredients in the cup
 - Add cup
 - Add hot water
 - Add coffee
 - Add creamer (if required)
 - Add sweetener (if required)
 - Heating the water
 - Switching heating element off
 - Heating the water
 - Switching heating element on
 - Obtaining ingredients
 - Obtaining resources (spoon, pot, etc.)
 - ...

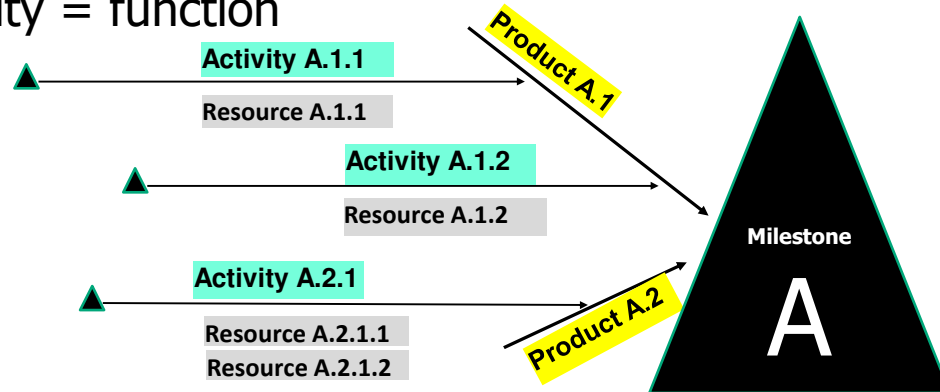
**Never do this,
always list
every item**

How to write good requirements

0401-22

Product-Activity-Milestone (PAM) chart*

- Activity = function

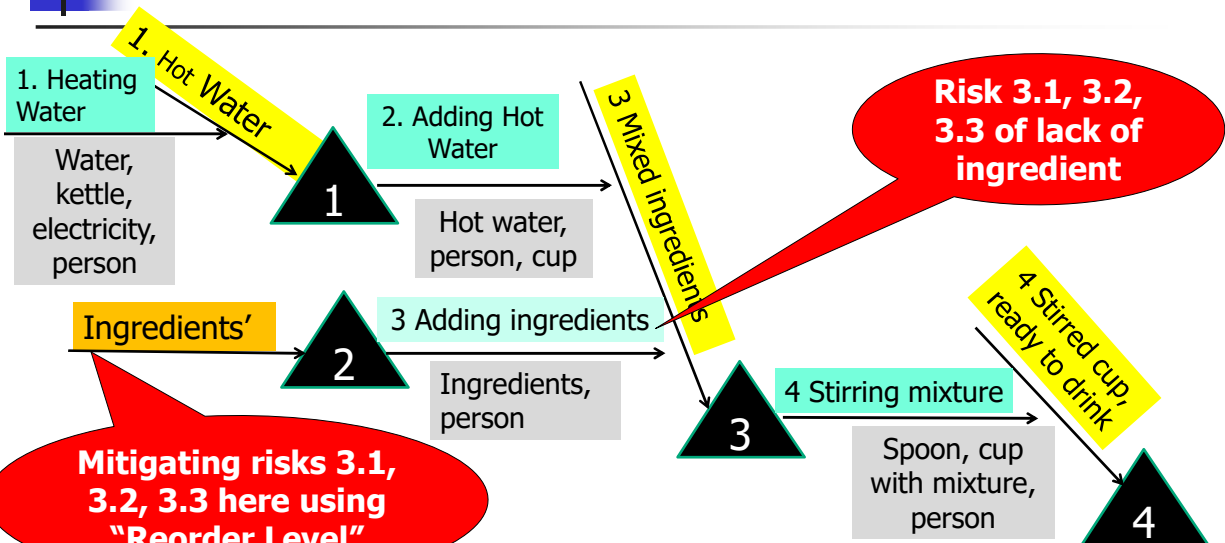


Systems Thinkers Toolbox 2.14

How to write good requirements

0401-23

Risk identification and mitigation/prevention (e.g.)



How to write good requirements

0401-24

Quantitative HTP: Risk likelihood (probability)

Degree of Probability	Description
Likely (5)	Likely to occur immediately or within a short period of time. Expected to occur frequently or continuously to an individual item or person.
Probably (4)	Probably will occur in time. Expected to occur several times to an individual team or person or frequently to a group
May (2)	May occur in time. Can reasonably be expected to occur some time to an individual item or person or frequently to a group
Unlikely (1)	Unlikely to occur

How to write good requirements

0401-25

Risk consequence (severity)

Category	Description
5	May cause death, loss of facility / asset or result in grave danger to mission
4	May cause severe injury, illness, property damage, damage to mission, or degradation to efficient use of assets
2	May cause minor injury, illness, property damage, degradation to efficient use of assets
1	Presents a minimal threat to personnel safety or health, property, mission or efficient use of assets

How to write good requirements

0401-26

Traditional Risk Assessment Matrix

Probability of occurrence (L) (Likelihood)	5	5	10	15	20	25
	4	4	8	12	16	20
	3	3	6	9	12	15
	2	2	4	6	8	10
	1	1	2	3	4	5
		1	2	3	4	5

Severity of consequences (S)
(Impact)

Based on one number ($L \times S$)

The level of risk for each root cause is reported as:

1-4	Low (green)
5-12	Moderate (yellow)
15-25	High (red)

How to write good requirements 0401-27

Where do numbers come from?

- Experience
- Historical data
 - E.g., actuarial data
- Expert estimates
- Standards
- Assumptions
- Guesses
- Other places

How to write good requirements 0401-28

Opportunities and risks (DOD EPMC course)*

		Opportunity				Risk				
Probability	NEARLY CERTAIN									
	HIGHLY LIKELY									
	LIKELY									
	UNLIKELY									
	REMOTE									
		GREAT	HIGH	FAIR	SLIGHT	NEUTRAL	MINOR	MODERATE	SERIOUS	CRITICAL



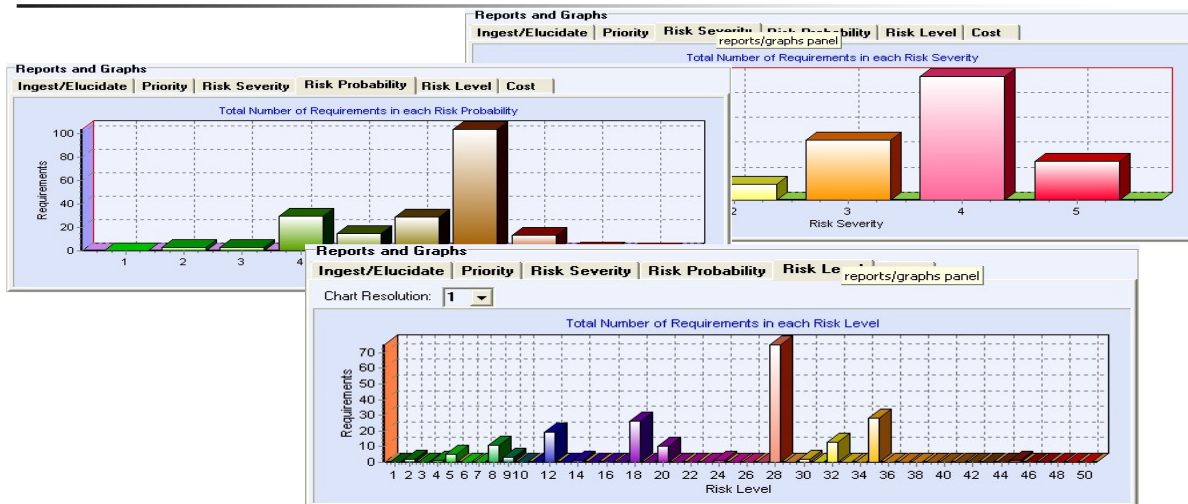
Red & Blue ratings justify major investment of resources & effort
 Yellow & White ratings justify secondary consideration
 Green ratings are low priority (monitor but leave alone)
 Probability rating = odds of achieving opportunity or risk occurring

* US DOD, 2004

How to write good requirements

4-29

Project Risk profiles*



How to write good requirements

0401-30



Exercise 4-11 scenarios to requirements

1. Update the extracted scenarios in your Exercise 2-11 presentation
2. Assign an identification to each scenario
3. Add at least one misuse function to at least one scenario
4. Write at least 10 well-written requirements with traceability to any of the scenarios including the scenario(s) updated with the misuse function(s)
5. Prepare a <5 minute presentation containing
 1. The misuse function in its scenario highlighting the modification(s) if any
 2. At least 10 well-written requirements including some traceable to the misuse function
 3. Traceability of each requirement to the source scenario and function
 4. A compliance matrix for the exercise
 5. The exercise problem formulated per COPS problem formulation template
 6. Lessons learned from exercise
 7. This slide and the version number of the session
6. Save as a PowerPoint file in format Exercise4.11-abcd.pptx
7. Post/email presentation as and where instructed

How to write good requirements

0401-31



Exercise 4-12 knowledge reading

1. Prepare a brief on two main points on reading 0402 (< 5min)
2. Presentation to contain
 1. Formulated problem per COPS problem formulation template
 2. A summary of the content of the reading (<1 minute)
 3. The compliance matrix
 4. This slide and the version number of the session
 5. The main points
 6. The two briefings
 7. Reflections and comments on reading (<2 minute)
 8. Comparisons of content with other readings and external knowledge
 9. Why you think the reading was assigned to the module
 10. Lessons learned from module and source of learning e.g. readings, exercise, experience, etc. (<2 minutes)
3. Save as a PowerPoint file as Exercise4-12-abcd.pptx
4. Post/email presentation as and where instructed
5. Brief on one main point

How to write good requirements

0201-32



Any questions ?

1. Best
2. Worst
3. Missing



Email: beyondsystemsthinking@yahoo.com

Subject: <class title> BWM Module #

How to write good requirements

0401-33