Evolutionary Requirements Engineering

Allen Byrne
CSDP 2003

Overview

- Problems with Requirements
- Process to Identify Requirements
- Examples

The Evolutionary Process

- The ‘evolutionary’ concept implies association with Evolutionary Project Management; an iterative process of change, feedback, learning and consequent change. Evolutionary processes need to be carefully distinguished from other processes – those that do not iterate, do not learn from experience, and do not cater for change.

---- Tom Gilb

Evolutionary Project Management

- **A project management process delivering ‘high-value-first’ progress towards the <u>currently</u> *defined and approved requirements*, and then seeking to obtain, and use, realistic, early feedback.**

Requirement Problems

- Using a language like English to communicate ideas precisely is challenging. English is simply not very precise. It leaves a lot of room for interpretations.
- Requirements are repeated in several places in similar but different ways.
- No clear idea of what are Stakeholder Values & Product Qualities

Stakeholder Value & Product Qualities

- **Definitions**
- Relationships
- The 7 Whys ? and 1 step back
- Justification

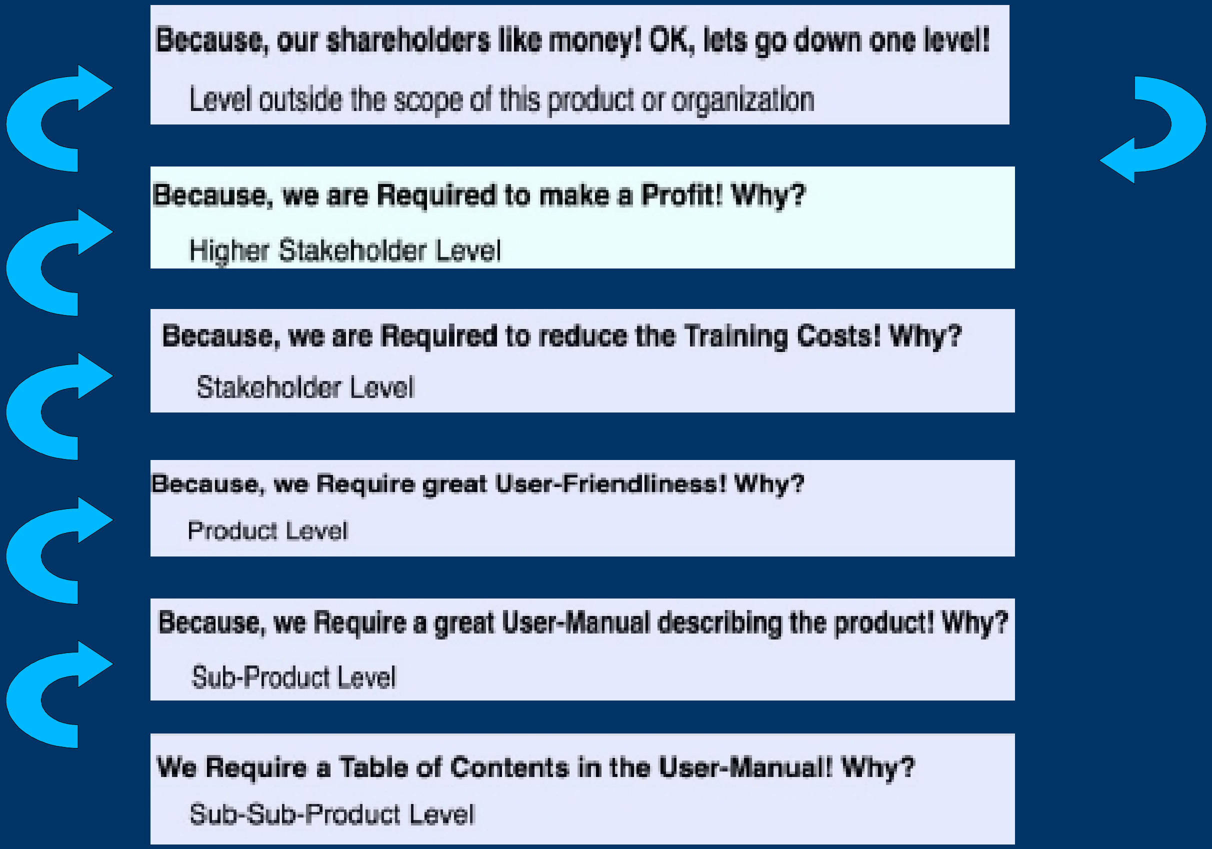
Stakeholder Value & Product Qualities

- Definitions
- **Relationships**
- The 7 Whys ? and 1 step back
- Justification

Stakeholder Value & Product Qualities

- Definitions
- Relationships
- **The 7 Whys ? and 1 step back**
- Justification

Why? Illustration



Because, our shareholders like money! OK, lets go down one level!

Level outside the scope of this product or organization

Because, we are Required to make a Profit! Why?

Higher Stakeholder Level

Because, we are Required to reduce the Training Costs! Why?

Stakeholder Level

Because, we Require great User-Friendliness! Why?

Product Level

Because, we Require a great User-Manual describing the product! Why?

Sub-Product Level

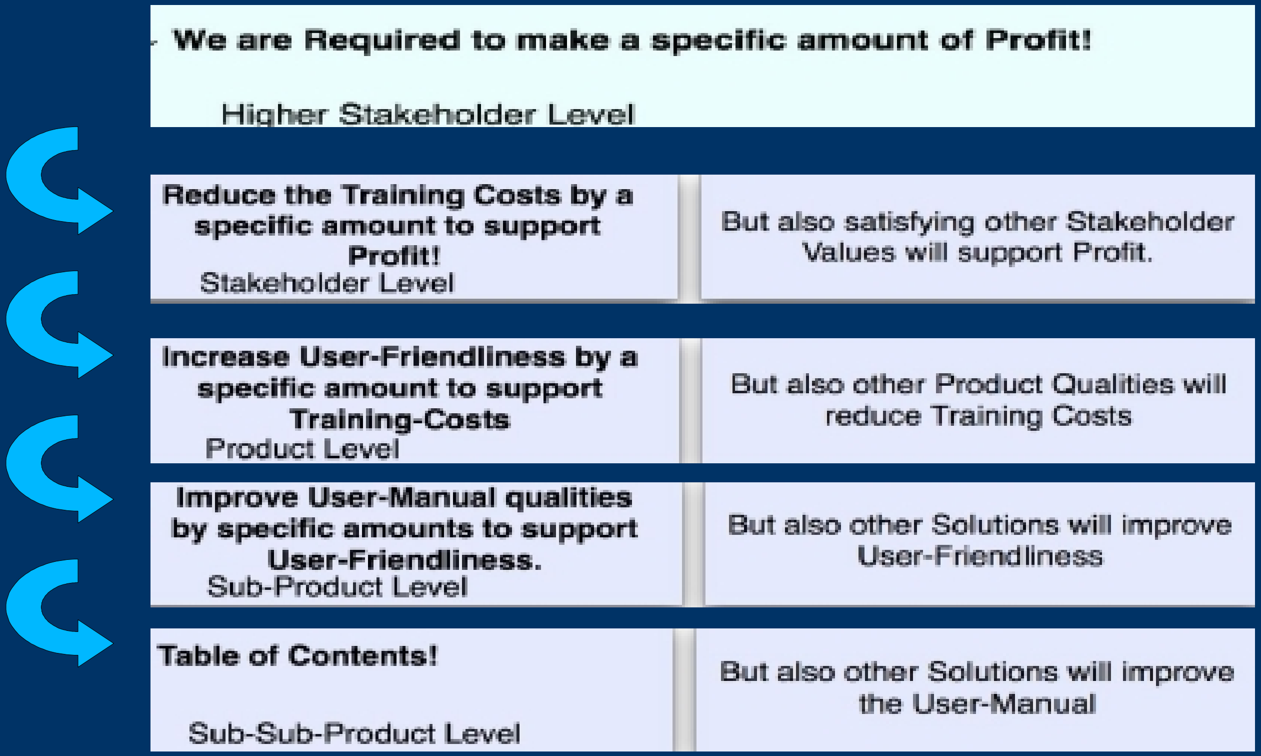
We Require a Table of Contents in the User-Manual! Why?

Sub-Sub-Product Level

Stakeholder Value & Product Qualities

- Definitions
- Relationships
- The 7 Whys ? and 1 step back
- **Justification**

How? Illustration



We are Required to make a specific amount of Profit!

Higher Stakeholder Level

Reduce the Training Costs by a specific amount to support Profit! Stakeholder Level	But also satisfying other Stakeholder Values will support Profit.
Increase User-Friendliness by a specific amount to support Training-Costs Product Level	But also other Product Qualities will reduce Training Costs
Improve User-Manual qualities by specific amounts to support User-Friendliness. Sub-Product Level	But also other Solutions will improve User-Friendliness
Table of Contents! Sub-Sub-Product Level	But also other Solutions will improve the User-Manual

Planning Language

- Planguage© Tom Gilb, is a specification language and a set of related methods for systems engineering.
- Planguage is designed to express ideas about requirements, designs and plans.
- A powerful yet simple tool to assure clarity, is used to define words and acronyms thoroughly.

Planguage Specification

- A set of defined concepts
- A set of defined parameters and grammar
- A set of icons

Process Specification

- Entry Conditions
- Procedure
- Exit Conditions

Requirement Specification

Procedure

P1: Define the system scope and the overall scope of the requirements.

P2: Identify relevant (critical and profitable) stakeholders.

P3: Determine the requirements of each type of stakeholder. Ensure all specification statements are source-referenced.

P4: Categorize requirements by type (the major requirement types are function requirement, performance requirement, resource requirement, design constraint and condition constraint).

P5: Specify Function Requirements (Process.FR. See Chapter 3).

P6: Specify Performance Requirements (Process.PR. See Chapter 4) including identifying or creating a Scale of Measure (Process.SD. See Chapter 5).

P7: Specify Resource Requirements (Process RR. See Chapter 6).

P8: Identify and question any design constraints and condition constraints.

(Are they real or was something else intended?) Ensure the necessary design and condition constraints are specified.

P9: Specify all known significant relationships of the requirements to any other relevant requirement specifications (external or internal to the system). You need to identify where there may be overlap or conflict or double accounting over benefits. There may even be synergy or a chance to 'subcontract' parts of the system development.

P10: Get stakeholders to approve the written requirement specifications that specifically affect them.

P11: Carry out Specification Quality Control (SQC) on the requirement specification. Obtain management review approval.

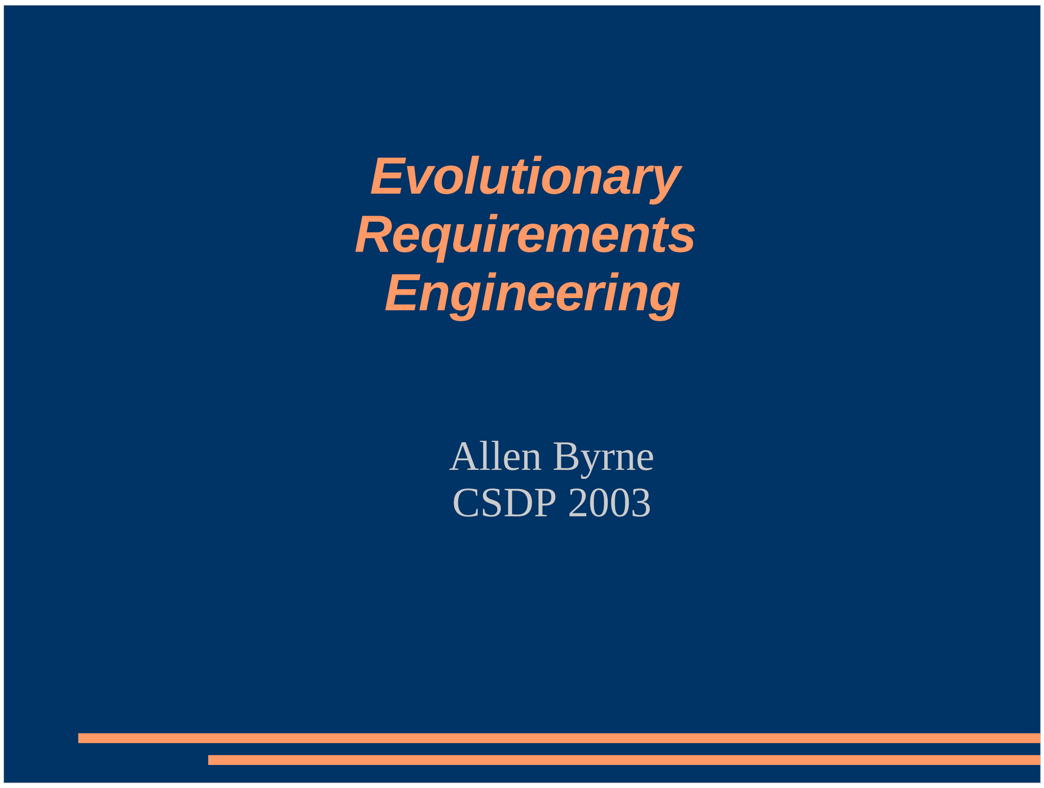
Requirement Specification Template (A Summary Template)
Tag: <Tag name for the system>.
Type: System.
========================== Basic Information ==========================
Version: <Date or other version number>.
Status: <{Draft, SQC Exited, Approved, Rejected}>.
Quality Level: <Maximum remaining major defects/page, sample size, date>.
Owner: <Role/e-mail/name of the person responsible for changes and updates>.
Stakeholders: <Name any stakeholders (other than the Owner) with an interest in the system>.
Gist: <A brief description of the system>.
Description: <A full description of the system>.
Vision: <The overall aims and direction for the system>.
============================ Relationships ============================
Consists Of: Sub-System: <Tags for the immediate hierarchical sub-systems, if any, comprising this system>.
Linked To: <Other systems or programs that this system interfaces with>.
======================== Function Requirements ========================
Mission: <Mission statement or tag of the mission statement>.
Function Requirement:
<{Function Target, Function Constraint}>: <State tags of the function requirements>.
Note: 1. See Function Specification Template. 2. By default, ‘Function Requirement’ means ‘Function Target’.
====================== Performance Requirements ======================
Performance Requirement:
<{Quality, Resource Saving, Workload Capacity}>: <State tags of the performance requirements>.
Note: See Scalar Requirement Template.
======================== Resource Requirements ========================
Resource Requirement:
<{Financial Resource, Time Resource, Headcount Resource, others}>: <State tags of the resource requirements>.
Note: See Scalar Requirement Template.
========================== Design Constraints ==========================
Design Constraint: <State tags of any relevant design constraints>.
Note: See Design Specification Template.
======================== Condition Constraints ========================
Condition Constraint: <State tags of any relevant condition constraints or specify a list of condition constraints>.
===================== Priority and Risk Management =====================
Rationale: <What are the reasons supporting these requirements? >.
Value: <State the overall stakeholder value associated with these requirements>.
Assumptions: <Any assumptions that have been made>.
Dependencies: <Using text or tags, name any major system dependencies>.
Risks: <List or refer to tags of any major risks that could cause delay or negative impacts to the achieving the requirements>.
Priority: <Are there any known overall priority requirements? >.
Issues: <Unresolved concerns or problems in the specification or the system>.
================== Evolutionary Project Management Plan ==================
Evo Plan: <State the tag of the Evo Plan>.
======================== Potential Design Ideas ========================
Design Ideas: <State tags of any suggested design ideas for this system, which are not in the Evo Plan>.

Requirement specification template.
This is a summary template giving an overview of the requirements.
<- CE Introduction to Requirements

============ Function Requirements ============
Mission: <Mission statement or tag of the mission statement>.
Function Requirement:
<{Function Target, Function Constraint}>: <State tags of the function requirements>.
Note: 1. See Function Specification Template. 2. By default, ‘Function Requirement’ means ‘Function Target’.
======== Performance Requirements =============
Performance Requirement:
<{Quality, Resource Saving, Workload Capacity}>: <State tags of the performance requirements>.
Note: See Scalar Requirement Template.
============ Resource Requirements ============
Resource Requirement:
<{Financial Resource, Time Resource, Headcount Resource, others}>: <State tags of the resource requirements>.
Note: See Scalar Requirement Template.
============= Design Constraints ==============
Design Constraint: <State tags of any relevant design constraints>.
Note: See Design Specification Template.
============== Condition Constraints =============
Condition Constraint: <State tags of any relevant condition constraints or specify a list of condition constraints>.

Summary

- There is a better way to develop products
- Identify all Stakeholders
- Clearly state and quantify Requirements
- Use the evolutionary process to deliver value to the stakeholders

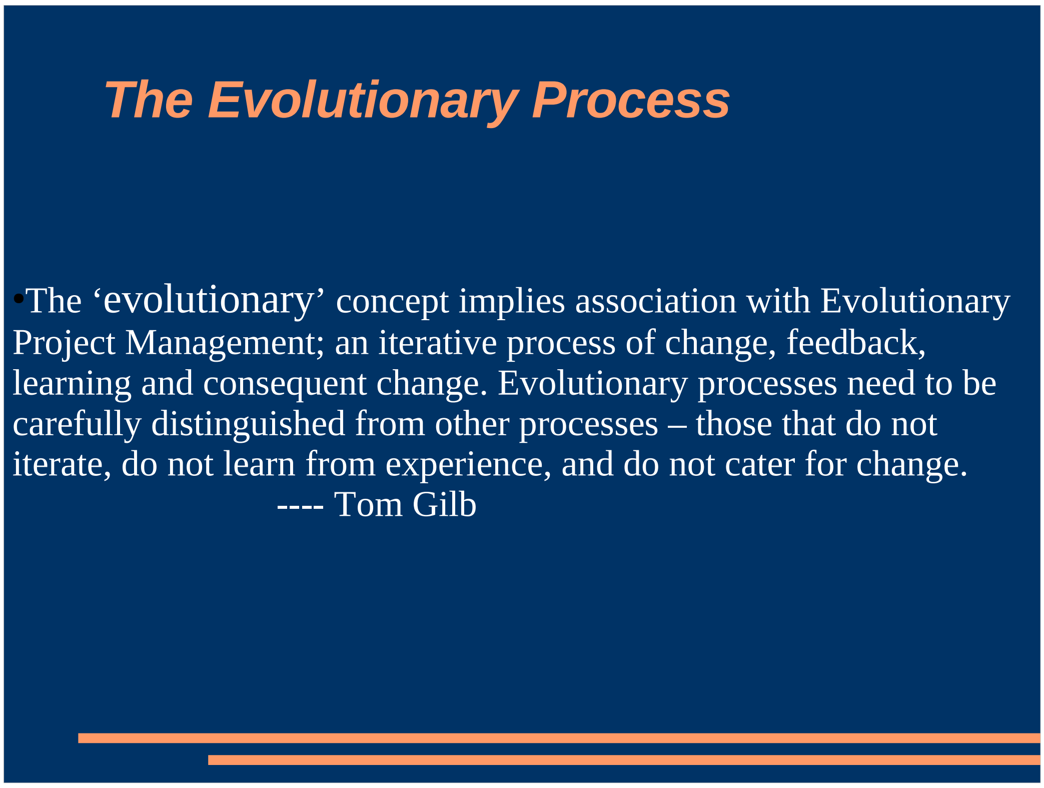


Evolutionary Requirements Engineering

Allen Byrne
CSDP 2003

Overview

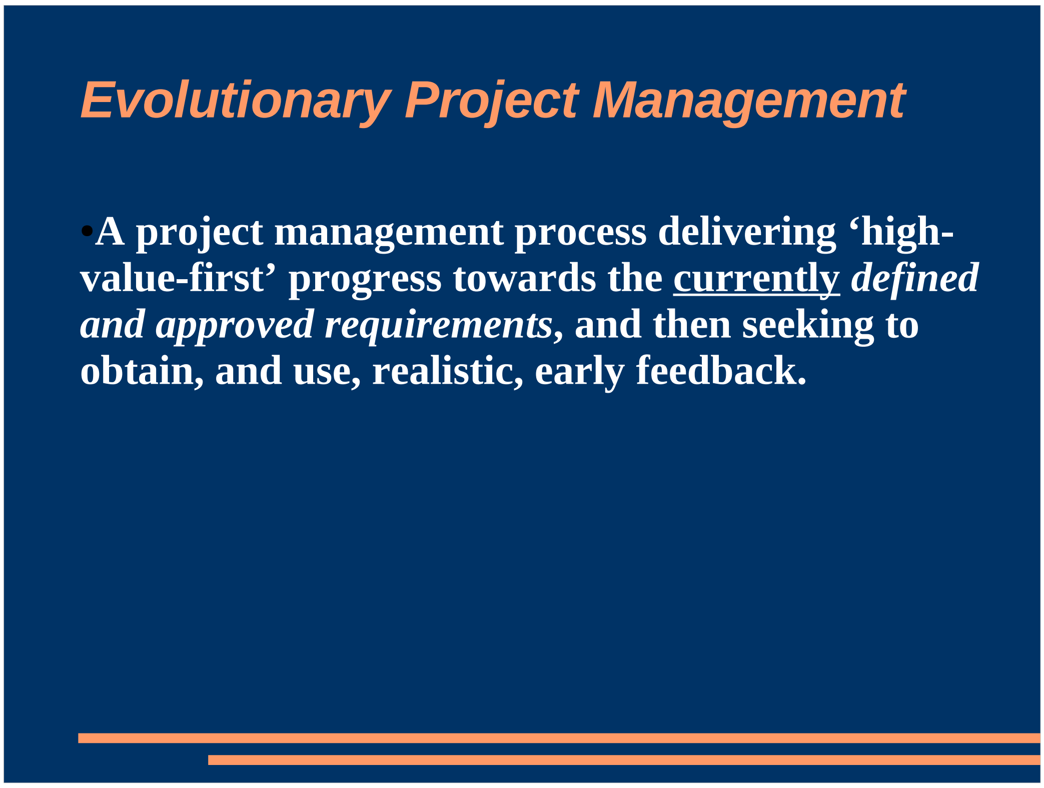
- Problems with Requirements
- Process to Identify Requirements
- Examples



The Evolutionary Process

•The ‘evolutionary’ concept implies association with Evolutionary Project Management; an iterative process of change, feedback, learning and consequent change. Evolutionary processes need to be carefully distinguished from other processes – those that do not iterate, do not learn from experience, and do not cater for change.
---- Tom Gilb

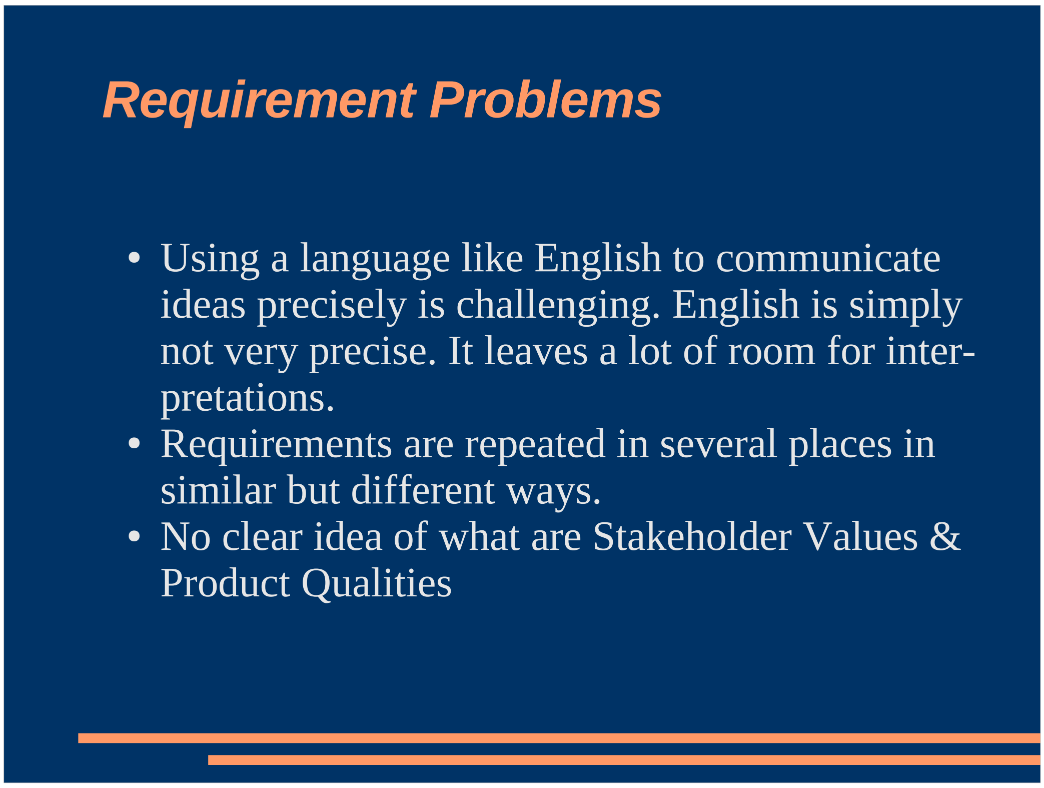
Tom Gilb is author of the book “Competitive Engineering” and creator of Planguage or the “Planning Language”. His ideas are the basis of many agile development techniques, and directly acknowledged by Kent Beck of “extreme programming” fame.



Evolutionary Project Management

•A project management process delivering 'high-value-first' progress towards the <u>currently</u> *defined and approved requirements*, and then seeking to obtain, and use, realistic, early feedback.

Key components include:
• frequent delivery of system changes (steps)
• steps delivered to stakeholders for real use
• feedback obtained from stakeholders to determine *next* step(s)
• the existing system is used as the initial system base
• small steps (ideally between 2%-5% of total project financial cost and time)
• steps with highest value and benefit-to-cost ratios given highest priority for delivery
• feedback used 'immediately' to modify future plans and requirements and, also to decide on the next step
• total systems approach ('anything that helps')
• results-orientation ('delivering the results' is prime concern)

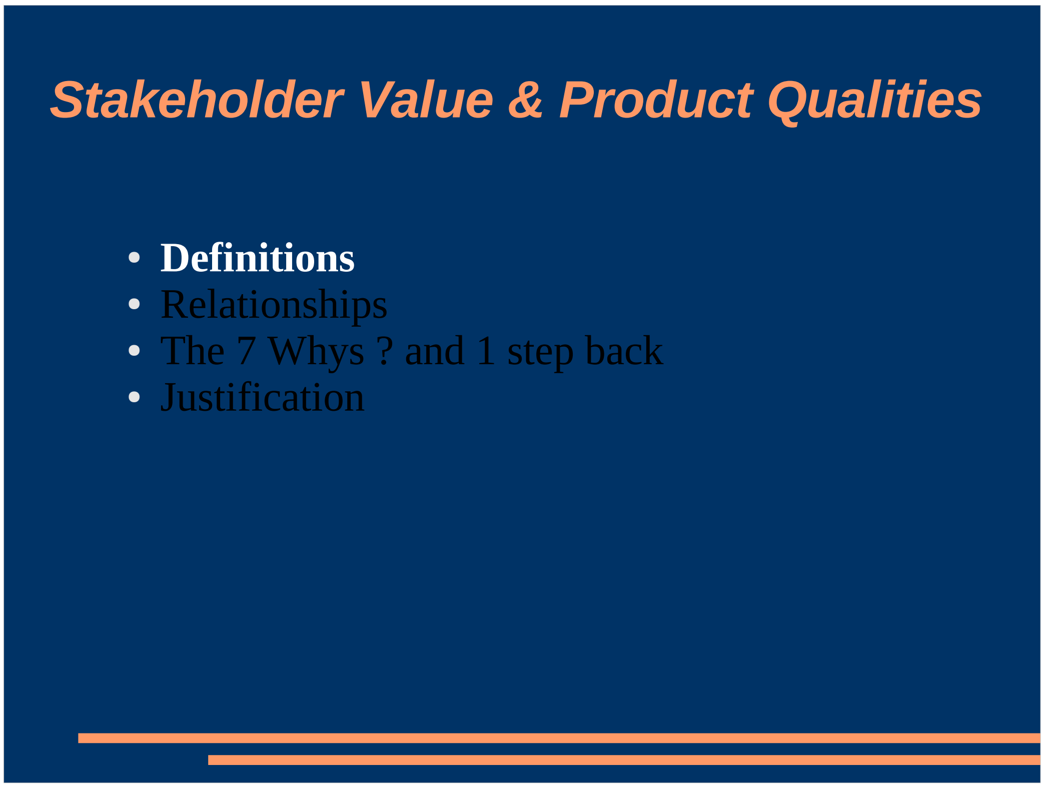


Requirement Problems

- Using a language like English to communicate ideas precisely is challenging. English is simply not very precise. It leaves a lot of room for interpretations.
- Requirements are repeated in several places in similar but different ways.
- No clear idea of what are Stakeholder Values & Product Qualities

Tom Gilb has over a period of several decades developed a method, a Planning Language or Planguage. He started developing and using Planguage in the late 1960. It has proven remarkably helpful in communicating ideas in projects clearly so everyone involved understands the ideas as they where intended. It also guides and stimulates the thinking process and creativity of the people writing the specifications. Planguage is very simple, yet very powerful. It is based on any language, like English, but it adds some structure and words with defined meanings. It is not trying to develop a fancy method, it is a reflection of reality, a way of describing products and projects.

One version of one requirement for the engineers, another version of the same requirement for the testing group, one for the marketing group and another version of the same requirement for the people writing the user manual. The justification for this practice seems to be that people have slightly different needs or something to do with readability. This practice is catastrophic and should be outlawed. It leads to variations in what is one and the same requirement. When a change needs to be done in one requirement, it is impractical to update them all. Usually it only gets updated in one instance of the requirement.



Stakeholder Value & Product Qualities

- **Definitions**
- Relationships
- The 7 Whys ? and 1 step back
- Justification

A requirement is a stakeholder need that a developer is planning to satisfy.

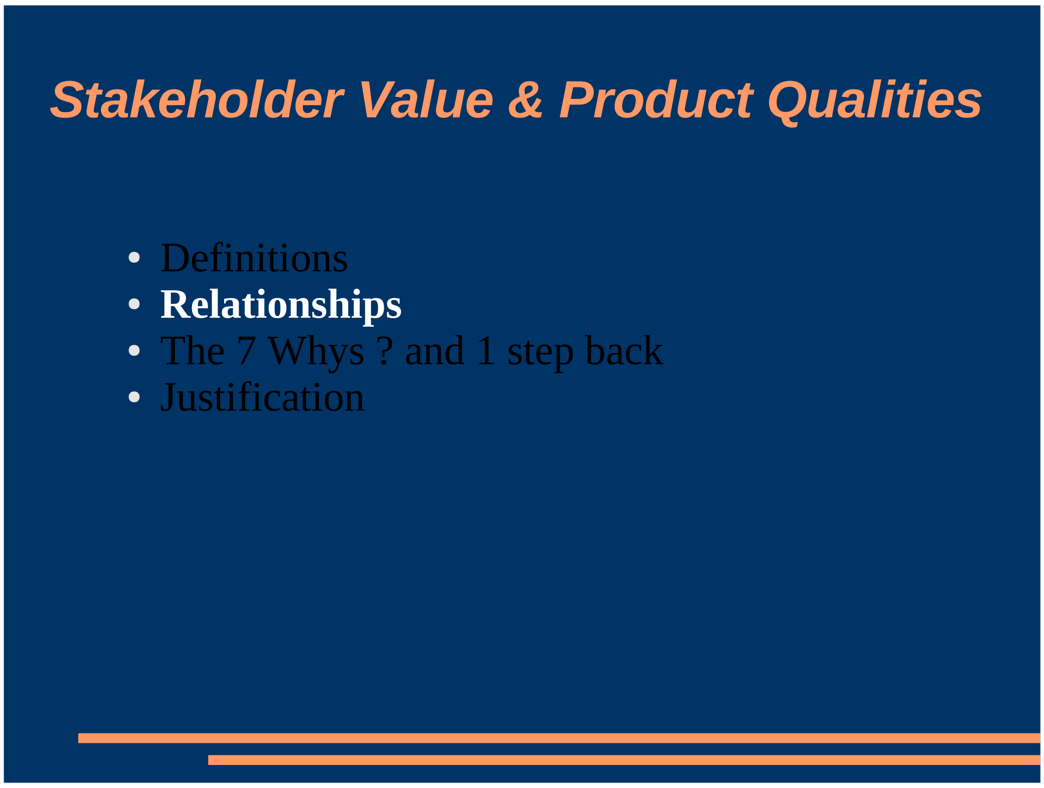
A stakeholder is any person, group or object, which has some direct or indirect interest in a system.

Value is a *perceived* benefit: that is, the benefit we think we get from something.

Stakeholder Values are the improvements the Stakeholder needs or desires for themselves, with or without the product. Notice that Stakeholder Values are product independent and can be fulfilled with any combination of products.

Product Qualities are how well a product interacts with people, other products and internally within the product.

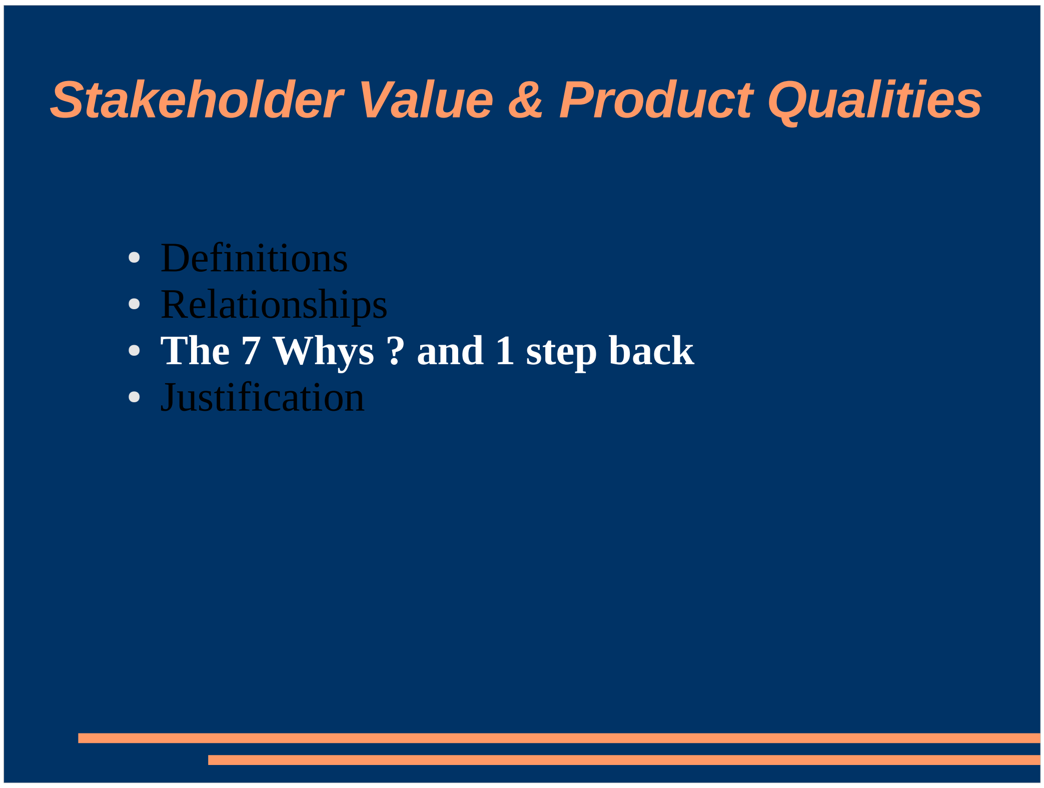
Development Resources can be anything which is in limited supply, and can be use to develop the Solutions to achieve a Stakeholder Value and Product Quality Requirement. Typically, time, qualified people, money, space.



Stakeholder Value & Product Qualities

- Definitions
- **Relationships**
- The 7 Whys ? and 1 step back
- Justification

Development Resources are used to run the development process. The Development Process develops new improved Solutions with enhanced Product Qualities. When the Stakeholder uses the new product with the enhanced Product Qualities, it improves on their Stakeholder Values.

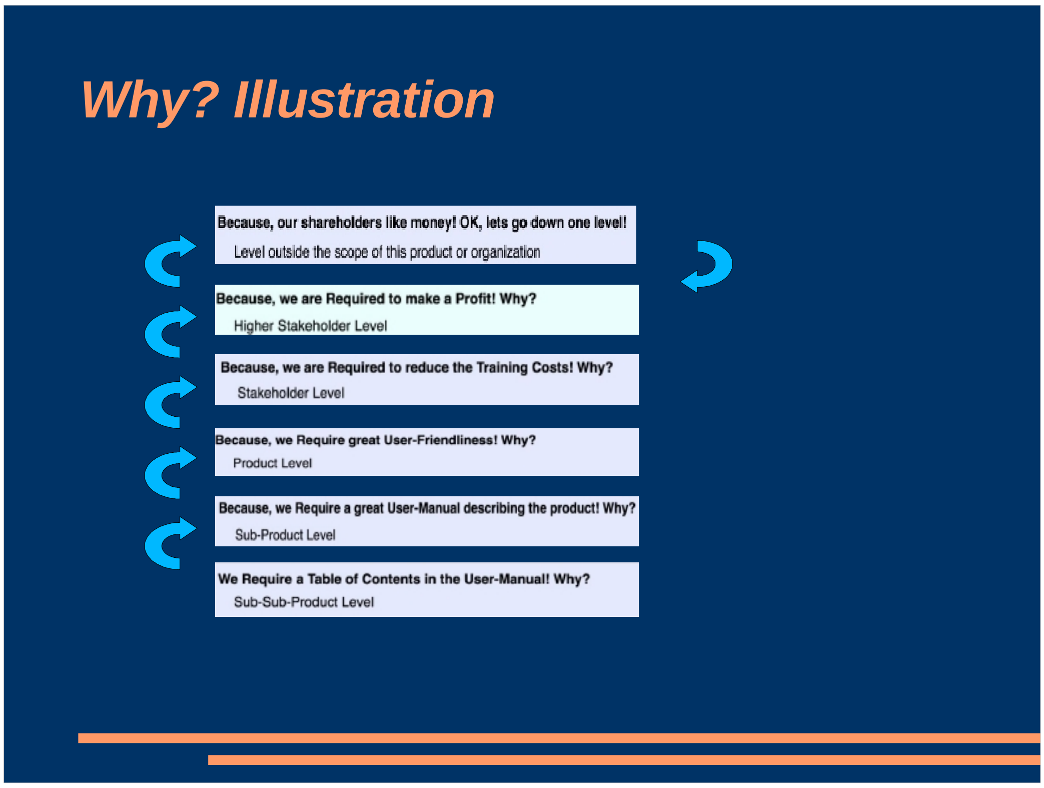


Stakeholder Value & Product Qualities

- Definitions
- Relationships
- **The 7 Whys ? and 1 step back**
- Justification

Stakeholder Value & Product Quality Requirements are something desired in the future, an end state. Solutions are means of getting there. There are always many possible potential Solutions to reach a set of Stakeholder Value & Product Quality Requirements. Many Solutions can satisfy the Requirements. A critical part of creating a competitive product or service is to find and use Solutions that are cheap and fast, yet satisfies the end state Requirements. If Solutions are in the Requirement specification, one can not pick the Solutions that best satisfies the Requirements. It is also highly likely that after developing and delivering Solutions based requirements that the real requirements will not be satisfied, and that the Stakeholders will be unhappy about the outcome. Finally, the Stakeholders will question the high cost in time and money.

To find the real Requirements, start a process of asking “why?“ 7 times. Each “why?”, finds a higher level Requirement. Moving from sub-product levels to product levels to Stakeholder Values, to higher level Stakeholder Values, to levels so high that it is not the concern of this project or organization, then stop and go one level back.



Why? Illustration

Because, our shareholders like money! OK, lets go down one level!
Level outside the scope of this product or organization

Because, we are Required to make a Profit! Why?
Higher Stakeholder Level

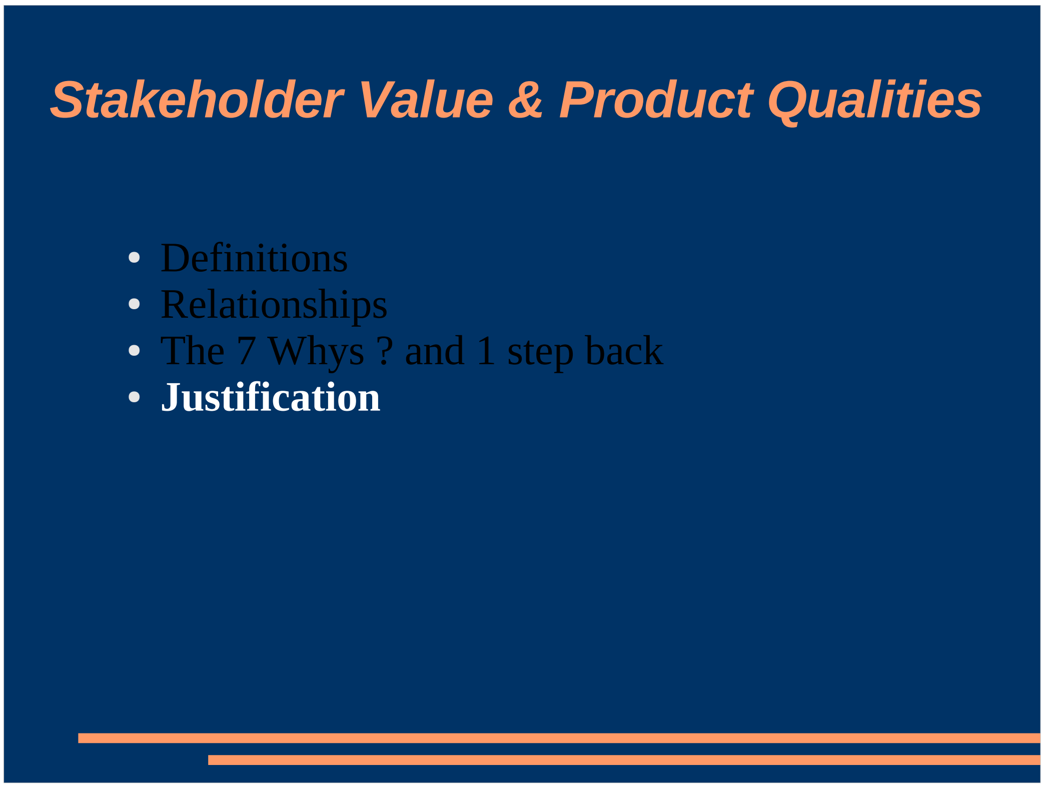
Because, we are Required to reduce the Training Costs! Why?
Stakeholder Level

Because, we Require great User-Friendliness! Why?
Product Level

Because, we Require a great User-Manual describing the product! Why?
Sub-Product Level

We Require a Table of Contents in the User-Manual! Why?
Sub-Sub-Product Level

Illustration: A Table of Contents in a User-Manual is written down as a Requirement. Many people would just list that as a valid Requirement, and start building a User-Manual with a Table of Contents. If we ask why?, why do we want a Table of Contents in our User-Manual? We see that a Table of Contents is a Solution that is there to support the idea of a great User-Manual. We move one level closer to the real need, the real Requirement. Then we continue this process of asking why?, and it turns out that a great User-Manual is also just a Solution for a Product Quality of User-Friendliness. Again we ask why do we want our product to be user-friendly?, and it turns out that User-Friendliness is there to help reduce the Training Cost which again is there to support the company in making a Profit. When we ask why do the company want to make a Profit?, we start getting answers that move outside the scope of the product and organization, and we can step down to Profit.



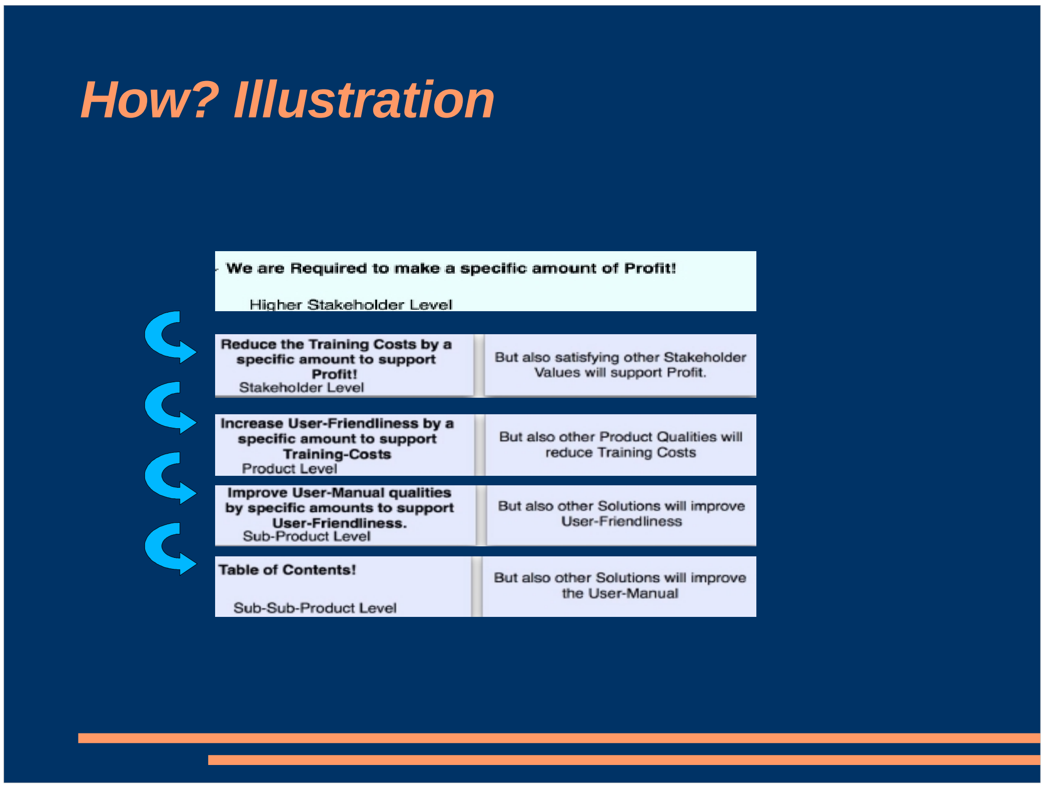
Stakeholder Value & Product Qualities

- Definitions
- Relationships
- The 7 Whys ? and 1 step back
- **Justification**

Viewed from the level preceding it, every level is a Solution, or set of Solutions. A level’s justification for existence is only the satisfaction of the level before it. Competitive advantage is achieved by designing each level so it best satisfies the level before it, with the minimum amount of Development Resources.

Through the process of asking “why?“, the actual reasons for what we are doing is revealed. We find the true Stakeholder Value & Product Quality Requirements. Only when we understand them can we make intelligent decisions about what Solutions are best suited to satisfy those Requirements.

Just because something is listed as a Requirement, does not make it so! Challenge it by asking why?

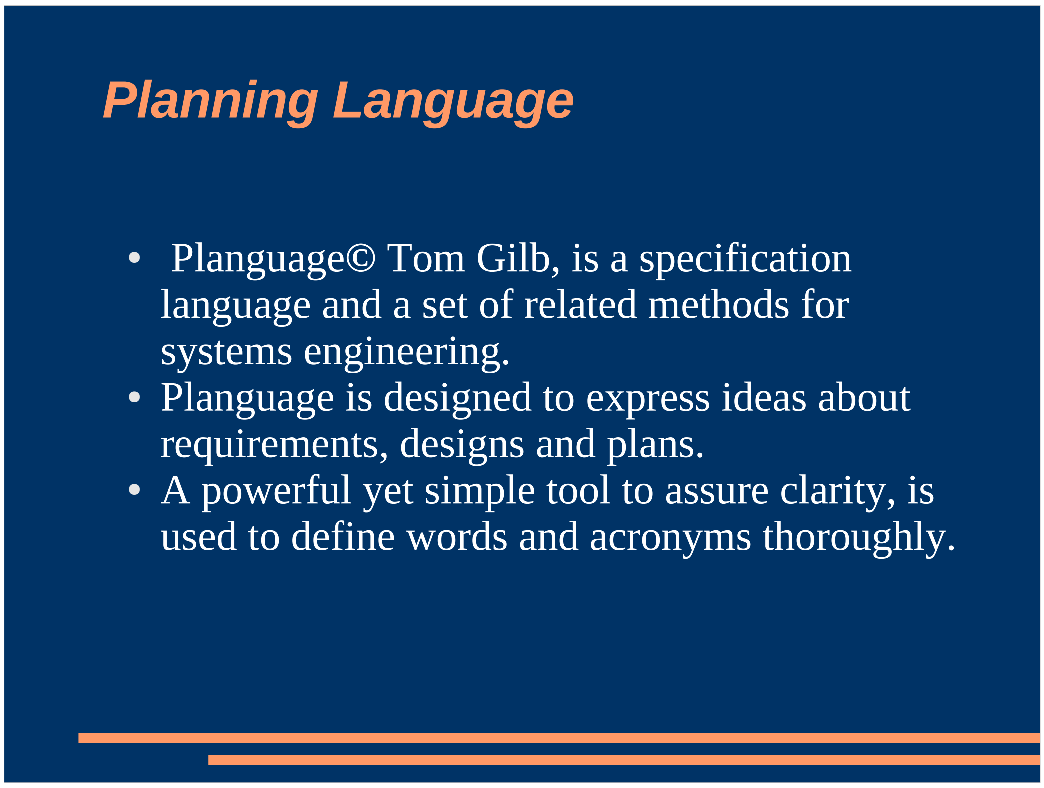


How? Illustration

We are Required to make a specific amount of Profit!
Higher Stakeholder Level

Reduce the Training Costs by a specific amount to support Profit! Stakeholder Level	But also satisfying other Stakeholder Values will support Profit.
Increase User-Friendliness by a specific amount to support Training-Costs Product Level	But also other Product Qualities will reduce Training Costs
Improve User-Manual qualities by specific amounts to support User-Friendliness. Sub-Product Level	But also other Solutions will improve User-Friendliness
Table of Contents! Sub-Sub-Product Level	But also other Solutions will improve the User-Manual

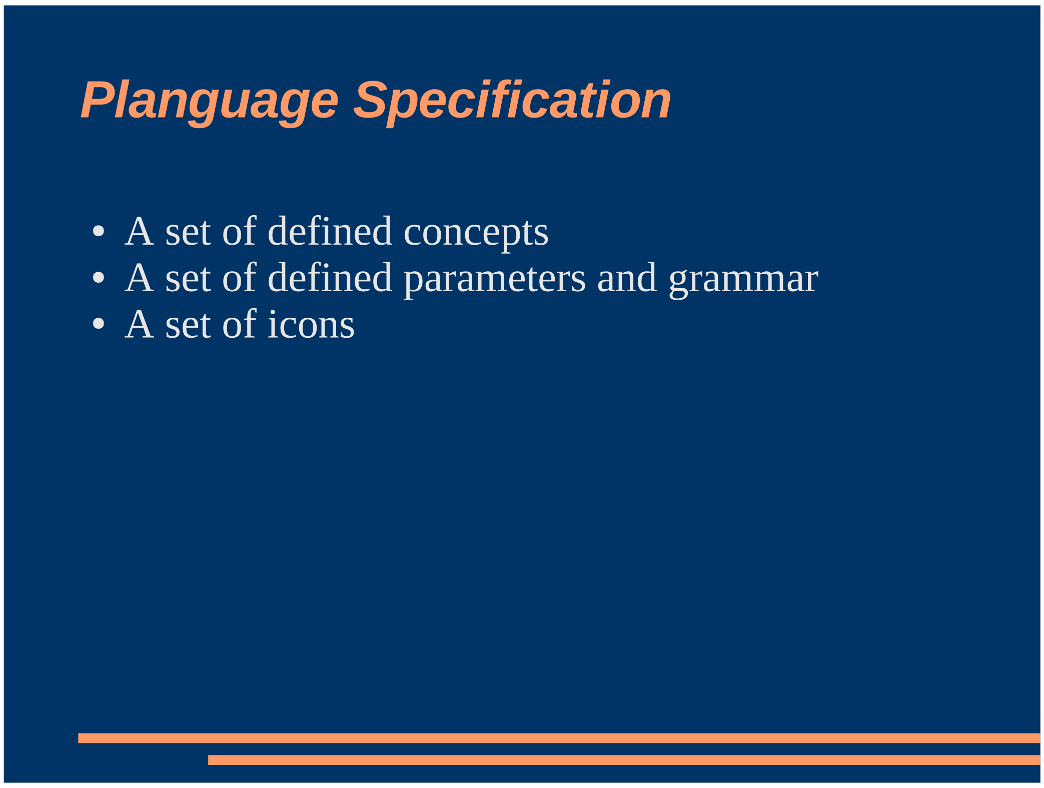
Illustration: Then we can continue the process in the other direction. How much Profit do we want? How much must we reduce the Training-Costs to get that Profit? What else must we do to get the Profit we want? To reduce the Training-Costs what can we do? Improve User-Friendliness, yes, but what else? What about; training the trainers better? Improving the Training Material? Home study training? Etc. To improve User-Friendliness, we can improve the User-Manual, but what else? What about; redesigning the User Interface so new users do not need to look things up in a User-Manual? What about designing a Mouse-Mat that has the answers to the 10 most common things looked up in the User-Manual. Etc. If we do want to have an improved User-Manual, how should the Table of Contents be for it to improve the User-Friendliness, and what other than a Table of Contents can we do improve the User-Friendliness?



Planning Language

- Planguage© Tom Gilb, is a specification language and a set of related methods for systems engineering.
- Planguage is designed to express ideas about requirements, designs and plans.
- A powerful yet simple tool to assure clarity, is used to define words and acronyms thoroughly.

All statements, whether they are part of Stakeholder Values, Product Qualities, Functions, Sub-Functions, Solutions, etc. will be useful only to the degree that they are unambiguous and clear to the intended reader intended by the author.

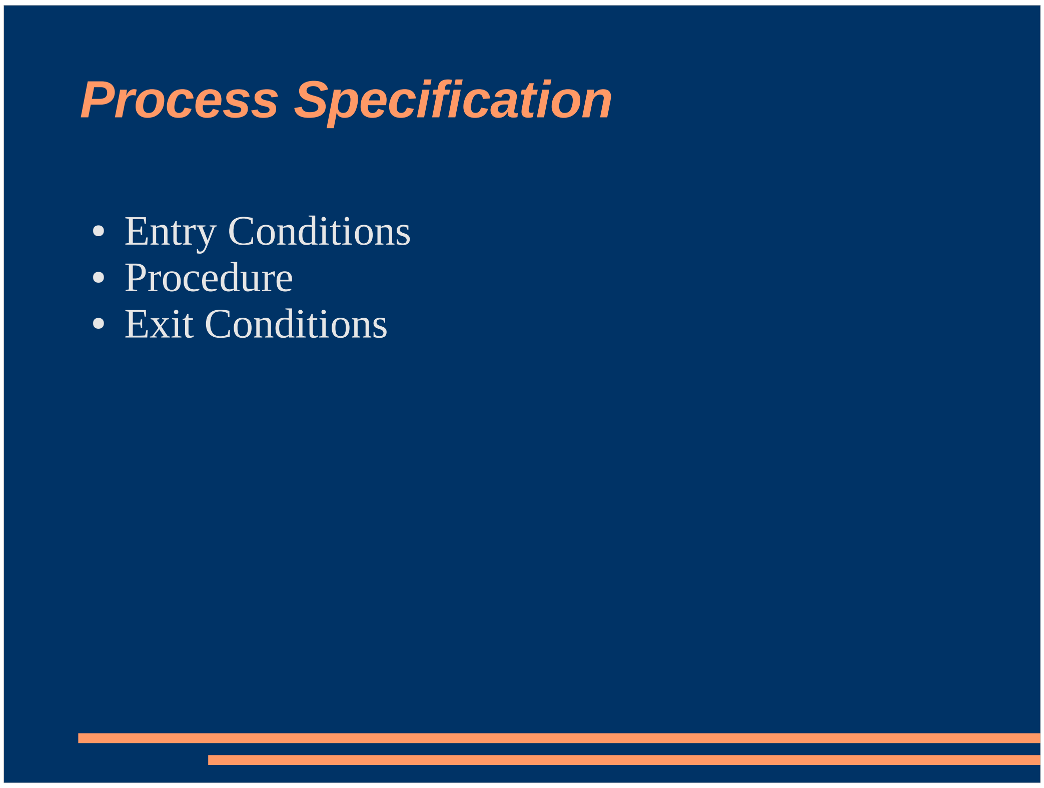


Planguage Specification

- A set of defined concepts
- A set of defined parameters and grammar
- A set of icons

The specification language (usually called simply ‘Planguage’) is used to specify requirements, designs and project plans. Planguage consists of the following elements:

The Planguage process descriptions (or methods) provide a recommended best practice for carrying out certain tasks.



Process Specification

- Entry Conditions
- Procedure
- Exit Conditions

Process Descriptions (or methods) are standards that describe the best practice for carrying out work tasks. The process format used for Planguage process descriptions consists of three basic elements:
 Entry Conditions – to determine whether it is wise to start the procedure
 Procedure – specifying for a task what work needs to be done and how best to do it
 Exit Conditions – to help determine if the work is ‘truly finished’.

Requirement Specification

Procedure

P1: Define the system scope and the overall scope of the requirements.

P2: Identify relevant (critical and profitable) stakeholders.

P3: Determine the requirements of each type of stakeholder. Ensure all specification statements are source-referenced.

P4: Categorize requirements by type (the major requirement types are function requirement, performance requirement, resource requirement, design constraint and condition constraint).

P5: Specify Function Requirements (Process.FR. See Chapter 3).

P6: Specify Performance Requirements (Process.PR. See Chapter 4) including identifying or creating a Scale of Measure (Process.SD. See Chapter 5).

P7: Specify Resource Requirements (Process RR. See Chapter 6).

P8: Identify and question any design constraints and condition constraints.

(Are they real or was something else intended?) Ensure the necessary design and condition constraints are specified.

P9: Specify all known significant relationships of the requirements to any other relevant requirement specifications (external or internal to the system). You need to identify where there may be overlap or conflict or double accounting over benefits. There may even be synergy or a chance to ‘subcontract’ parts of the system development.

P10: Get stakeholders to approve the written requirement specifications that specifically affect them.

P11: Carry out Specification Quality Control (SQC) on the requirement specification. Obtain management review approval.

Requirement specification template.
This is a summary template giving an overview of the requirements.
<- CE Introduction to Requirements 77

Requirement Specification Template (A Summary Template)
Tag: <Tag name for the system>.
Type: System.
=========================== Basic Information
===========================
Version: <Date or other version number>.
Status: <{Draft, SQC Exited, Approved, Rejected}>.
Quality Level: <Maximum remaining major defects/page, sample size, date>.
Owner: <Role/e-mail/name of the person responsible for changes and updates>.
Stakeholders: <Name any stakeholders (other than the Owner) with an interest in the system>.
Gist: <A brief description of the system>.
Description: <A full description of the system>.
Vision: <The overall aims and direction for the system>.
============================= Relationships
=============================
Consists Of: Sub-System: <Tags for the immediate hierarchical sub-systems, if any, comprising this system>.
Linked To: <Other systems or programs that this system interfaces with>.
========================= Function Requirements
=========================
Mission: <Mission statement or tag of the mission statement>.
Function Requirement:
<{Function Target, Function Constraint}>: <State tags of the function requirements>.
Note: 1. See Function Specification Template. 2. By default, ‘Function Requirement’ means ‘Function Target’.
======================= Performance Requirements
=======================
Performance Requirement:
<{Quality, Resource Saving, Workload Capacity}>: <State tags of the performance requirements>.
Note: See Scalar Requirement Template.
========================= Resource Requirements
=========================
Resource Requirement:
<{Financial Resource, Time Resource, Headcount Resource, others}>: <State tags of the resource requirements>.
Note: See Scalar Requirement Template.
=========================== Design Constraints
===========================
Design Constraint: <State tags of any relevant design constraints>.
Note: See Design Specification Template.
========================= Condition Constraints
=========================
Condition Constraint: <State tags of any relevant condition constraints or specify a list of condition constraints>.
====================== Priority and Risk Management
======================
Rationale: <What are the reasons supporting these requirements? >.
Value: <State the overall stakeholder value associated with these requirements>.
Assumptions: <Any assumptions that have been made>.
Dependencies: <Using text or tags, name any major system dependencies>.
Risks: <List or refer to tags of any major risks that could cause delay or negative impacts to the achieving the requirements>.
Priority: <Are there any known overall priority requirements? >.
Issues: <Unresolved concerns or problems in the specification or the system>.
=================== Evolutionary Project Management Plan
===================
Evo Plan: <State the tag of the Evo Plan>.
========================= Potential Design Ideas
=========================
Design Ideas: <State tags of any suggested design ideas for this system, which are not in the Evo Plan>.

============ Function Requirements
===========
Mission: <Mission statement or tag of the mission statement>.
Function Requirement:
<{Function Target, Function Constraint}>: <State tags of the function requirements>.
Note: 1. See Function Specification Template. 2. By default, ‘Function Requirement’ means ‘Function Target’.
======== Performance Requirements
=============
Performance Requirement:
<{Quality, Resource Saving, Workload Capacity}>: <State tags of the performance requirements>.
Note: See Scalar Requirement Template.
============ Resource Requirements
============
Resource Requirement:
<{Financial Resource, Time Resource, Headcount Resource, others}>: <State tags of the resource requirements>.
Note: See Scalar Requirement Template.
============= Design Constraints
==============
Design Constraint: <State tags of any relevant design constraints>.
Note: See Design Specification Template.
============== Condition Constraints
=============
Condition Constraint: <State tags of any relevant condition constraints or specify a list of condition constraints>.

Summary

- There is a better way to develop products
- Identify all Stakeholders
- Clearly state and quantify Requirements
- Use the evolutionary process to deliver value to the stakeholders