

Chapter 12



Prof. Dr. Harald Störrle
Danmarks Tekniske Universitet (DTU)

Chapter 12: Requirements Management

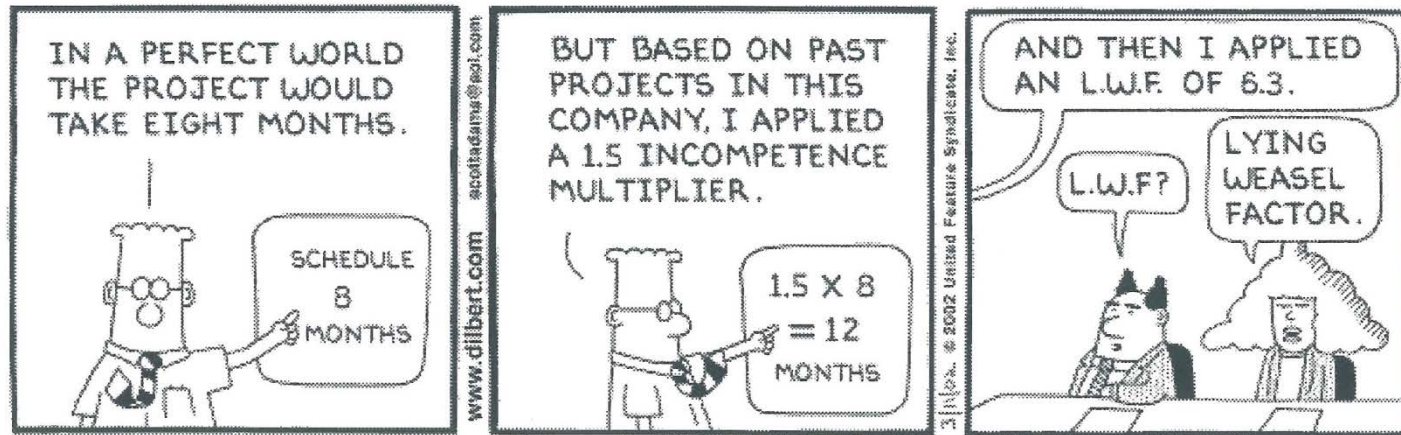
DTU course 02264

Abstract

- Requirements Management encompasses the overall process of requirements engineering, with a focus on administrative rather than engineering activities. This includes all tasks needed to assess, prioritize, and track requirements, but extends also to project management based on requirements engineering artifacts, and version control of requirements.
- We conclude this chapter with some examples of tools for RE.

Contents

1. Effort Estimation
2. Prioritization
3. Project Control with Requirements
4. Requirements Management Tools
5. Tailoring the Requirements Toolbox



Prof. Dr. Harald Störrle
Danmarks Tekniske Universitet (DTU)

Chapter 12.1

Effort Estimation

DTU course 02264

Effort Estimation

- **Clients and managers are oriented towards a project bottom-line, thus they are frequently inclined to ask questions such as:**
 - How much will it cost?
 - How long will it take?
- **The actual answers to these questions can of course only ever be given after completion, but an estimate is often sufficient.**
 - If the development organisation has done many similar projects before, it is relatively easy to arrive at an accurate estimate based on previous experience. This applies, e.g., to many maintenance tasks.
- **The quality of an estimate is determined by the following factors.**
 - Accuracy: How close is the estimate to the actual value.
 - Effort: what effort does it take to produce the estimate
 - Timeliness: how fast can the estimate be delivered.
- **The extreme cases are perfect accuracy (can be delivered after completion), and minimal cost/preparation time (“shoot from the hip”).**

Real-Life Estimation Methods

- **There are three ways of arriving at an estimate that are commonly practiced, each with their own advantages and issues.**
- **Expert estimation: a single expert gets to estimate the effort/cost.**
 - Obviously, the accuracy and reliability critically depends on the expert and his/her domain knowledge and experience.
 - Fast and simple, but unreliable.
- **Delphi-Estimation: a set of experts each deliver their own estimation anonymously, then may correct in the face of deviating estimates.**
 - Relatively reliable, but slow.
- **Group Estimation: a set of experts sit together to estimate various factors and discuss deviations.**
 - Information about the task is uniformly distributed, common awareness of risks is achieved.
 - There are many factors that can bias the results, such as group pressure, maturation, and ambition (if the estimators are likely to be involved later, too).

Estimation with Planning Poker

- **For very small features, a group dynamic exercise can be used for effort estimation in groups (popular in “agile” projects).**

- **Preparation**
 - The project leader rallies 4 to 6 developers, and deals “planning cards” to them. Each cards shows a number indicating some effort (usually days).

- **Estimation**
 - The project leader briefly presents the feature to be estimated and asks the team members for their estimates. Each one draws the appropriate card without showing them (yet).
 - The project leader gives a command, and everybody presents their estimation cards.
 - Those team members with the highest and lowest estimates are asked to justify their estimates. Then the team estimates again.
 - This process is repeated until all estimates are at least one card apart.

Improving Estimation Accuracy

- **The estimation accuracy relies on two factors.**
 - Clearly, the expertise and experience of those involved in the estimation is crucial, but it may not be representative for innovative projects.
 - Also, the degree of exploration of the task to be estimated can allow to increase accuracy and reliability at the price of more effort.
- **There are several methods that attempt to provide a structured way to explore the task such as to ensure higher estimation accuracy.**
 - The most well-known of these approaches is COCOMO (B. Boehm), that is, however, not widely used, maybe due to the effort involved.
- **Another, more pragmatic method, are so-called Use Case Points.**
 - Here, a number of properties of a future development project are elicited, estimated in isolation, and then combined.
 - The method produces highly reliable estimates at reasonable effort, but requires a degree of professionalism and dedication that is not always guaranteed.

Estimation with Use Case Points

- A UCP estimate is based on four factors, divided into **System Factors**

R Requirements factor: derived from estimating individual use cases and actors

M Management factor: derived from estimating 9 project properties

Project Factors

T Technology factor: correction factor for technological challenges

P Productivity factor: an experience value for a given organization

- These factors are then multiplied to form the overall effort estimation.

$$\text{Effort} = R * T * M * P$$

- **Pros and Cons**

- There is fairly good empirical support for UCP values, but the estimate depends largely on subjective assessments (R-, M-, T-factors), and previous measurements (P-factor).
- This support is mostly gathered for bespoke software, so it may not carry over to other kinds of development.

Use Case Points: R-Factor

- The **R** Requirements factor combines assessments of use cases and actors.

- In the simplest approach, the complexity of these may be estimated.

| Use Case Complexity | $weight(u)$ |
|---------------------|-------------|
| High | 15 |
| Medium | 10 |
| Low | 5 |

| Kind of Actor | $weight(a)$ |
|---------------|-------------|
| Human | 15 |
| Port | 10 |
| API | 5 |

- A more precise approach analyses the use case scenarios and interaction elements:

$$weight(u) = \sum_{s:scenario(u)} \frac{1}{3} \cdot steps(s) \qquad weight(a) = \frac{3}{4} interaction\ elements(a)$$

- The isolated assessments are summed up for all use cases and actors.

$$R = \sum_{u:usecase} weight(u) * \sum_{a:actor} weight(a)$$

- These approaches may be combined, e.g., counting ~30% of the use cases and actors, and assessing the remaining elements.

Use Case Points: M-Factor

- The **M**anagement factor combines the assessments of nine individual weighted factors as $M = \prod_{i=1..9} 1 + 0.1 \cdot W_i \cdot (3 - M_i)$
- The factors M_i (see below for details) assess a number of project properties related to
 - organizational maturity (i.e., the properties measured by SPICE/CMM(I) or similar approaches);
 - quality and stability of the specification (which might be determined by an inspection);
 - customer demands in terms of schedule/deadlines; and
 - context dependencies like the number of stakeholders, and the number of interfaces to be integrated.
- Some of these factors can be determined objectively, some require estimates or judgments.

Use Case Points: M-Factor (Sub aspects)

| i | M_i (Factor) | Ranges | W_i |
|---|--------------------------------|--|-------|
| 1 | Capabilities of Chief Designer | Chief Designer played this role in similar projects before: not yet (0); once (3); twice or more often (5) | 1.4 |
| 2 | Teamwork | How well does the overall team work together: bad (0); satisfactory (3); excellent (5) | 0.0 |
| 3 | Staff Continuity | How large is the continuity in the project team (% per year): >50% (0); ~25% (3); <10% (5) | 0.3 |
| 4 | Specification Quality | How many issues are there with the specification: many (0); some (3); few (5) | 0.5 |
| 5 | Process Maturity | How high is the complexity of the development process: high (0); normal (3); low (5) | 1.5 |
| 6 | Deadline Pressure | How much speed-up over the optimal duration does the client demand? more than 20% (0); about 10% (3); none (5) | 0.0 |
| 7 | Requirements Stability | How much change in the requirements is expected: very much (0); normal (3); little (5) | 1.8 |
| 8 | Number of Stakeholders | How many important stakeholders must be satisfied: >15 (0); 6-15 (3); 1-5 (5) | 0.0 |
| 9 | System Integration Challenges | How many new interfaces in existing systems are needed: >12 (0); 5-12 (3); 0-4 (5) | 0.7 |

Use Case Points: T- and P-Factors

- The **T**echnology factor measures the technological complexits of the development project, which is mainly determined by the required quality attributes (non-functional requirements).
- The „normal“ complexity of an information systems development project yields a T-factor of 1, but the published material on the UCP-estimation method is somewhat vague here.
 - Higher complexity (more and/or more stringent quality attributes) result in a higher T-factor.
- The **P**roductivity factor measures the productivity of the organization which will do the development.
- It can only be gathered from measuring the productivity in previous projects of similar kind, and with similar methods and organizations.
 - The P-factor usually ranges between 20—40 h/UCP.

Coarse Estimates with UCP

- Analyzing 16 past projects produced estimates for average weight factors for use cases and actors.

| Project size [person-days] | U-weight [average] | Variation [%] | A-weight [average] | Variation [%] |
|-------------------------------|-----------------------|------------------|-----------------------|------------------|
| Large (>5000) | 9.5 | 37% | 13 | 9% |
| Medium (2500-5000) | 6 | 19% | 10 | 27% |
| Small (625-2500) | 7 | 26% | 10.5 | 15% |

- Based on these, we may provide a first rough estimate of project effort based only on the number of use cases and actors, without assessing them in detail.
 - Observe, that there are large variances, and that these are for relatively large projects for bespoke (i.e., custom-made) software.
 - One UCP corresponds to roughly 18.9 person-days.



Prof. Dr. Harald Störrle
Danmarks Tekniske Universitet (DTU)

Chapter 12.2

Prioritization

DTU course 02264

Requirements Arbitrage

- **In some situations, effort estimation is not possible or not needed.**
 - For instance, in exploratory projects where the scope of the system is not fixed at the outset, a complete estimation is not possible.
 - Similarly, when there are more requirements than the client can or wants to afford, we have to select a viable set.
 - Finally, even if there is an up-front cost estimation, it is usually impossible to address all requirements simultaneously.
- **Finding out which requirements to address and which to neglect (for now or permanently) is called requirements arbitrage.**
- **Simply asking the client to prioritize the requirements is not a promising approach.**
 - Given that the outcome of requirements arbitrage typically affects the stakeholders, arbitrage is a highly political process.
- **Thus, it is very important to provide objective criteria and techniques to compare requirements and find reasons to justify a prioritization.**

Comparing Requirements by Satisfaction

- **Obviously, only very small sets of requirements are suitable for direct pairwise comparison.**
 - In the real world, the number of requirements (and thus the effort) will be too high, the comparisons may yield a circular structure, or the customer may not even be able to do many of the comparisons in the first place.
- **It is easier to ask clients for their assessment of the benefit/cost of the presence/absence of each requirement in isolation.**
 - Since benefit and cost come with an implication of objectivity and make it hard for cautious clients to commit to any judgment, it is easier to ask for a clearly subjective point of view, e.g., ask for satisfaction/dissatisfaction.
 - Observe, that satisfaction (when present) and dissatisfaction (when missing) are not two ends of one spectrum, but extremes on two different scales.
- **This results in a satisfaction matrix as proposed in the Volere approach.**

| | | Dissatisfaction (if missing) | |
|------------------------------|----|---------------------------------|----|
| | | hi | lo |
| Satisfaction (if present) | hi | A | C |
| | lo | B | D |

| | |
|---|---|
| A | Address |
| B | bread-and-butter requirements boring business parts or technology danger of not impressing customer |
| C | fancy functions danger of gold-plating |
| D | Ignore |

Comparing Requirements by Risk

- **One way of looking at requirements is by the risks they entail or reduce/avoid.**
 - Consider different safety requirements in the Therac-25 example. They are all important, but which are more important than others?
- **An objective way of answering that question could look at the risks they address: those requirements that eliminate greater risks should be preferred.**
 - In order to do that, we need to quantify the size of the risk.
 - Observe that size quantification is not necessarily done by associating risks with monetary cost, or indeed by any absolute assessment: a relative assessment is sufficient.

Risk Assessment

- **The assessment of risks is done in two steps.**
 - First, the magnitude of a risk event is assessed as a combination of the probability of its occurring, and the significance of the outcome.
 - Second, the magnitude is combined with the detectability of the event.
- **In order to maximize cost-effectiveness, events that are easier to detect are assigned higher priority.**

| Magnitude | | Loss | | |
|------------------------|--------|------|--------|-----|
| | | high | medium | low |
| Occurrence Probability | high | C | C | B |
| | medium | C | B | A |
| | low | B | A | A |

| Priority | | Detectability | | |
|-----------|---|---------------|--------|-----|
| | | high | medium | low |
| Magnitude | A | 2 | 1 | 1 |
| | B | 3 | 2 | 1 |
| | C | 3 | 3 | 2 |

Comparing Requirements by ROI

- Another way of prioritizing requirements might be by analyzing their return on invest (ROI, see below for a definition).
- When we quantify the ROI of all requirements, we can simply sort them by their value, and progress accordingly.
- As with risk assessment, the ROI assessment requires human judgment, so it is not entirely objective.
- Theoretically, risk assessment can be reduced to ROI analysis by assigning price tags to risks.
 - This may be acceptable for smaller size risks such as minor financial losses (e.g. credit card fraud), but is controversial for larger risks involving many human lives (e.g. nuclear energy).

Benefits and Costs of Requirements

Costs

- **Investment**
 - Initial development costs:
 - Cost of development team
 - Consultant fees
 - Tools, libraries, middleware
 - Hardware (which, buy/lease)
 - facilities (site, power,...)
 - Deployment costs:
 - installing the system,
 - training personnel, file conversion,....
- **Operational costs**
 - System Maintenance
 - hardware (repairs, lease, supplies,...),
 - software (licenses and contracts),
 - facilities
 - Personnel
 - For operation (data entry, backups,...)
 - For support (user support, HW/SW maintenance, supplies,...)
 - On-going training costs

Benefits

- **Tangible Benefits**
 - Readily quantified as money amounts, e.g.:
 - increased sales, increased margin on sales
 - reduced cost/errors
 - increased throughput/efficiency
 - more effective use of staff time
- **Intangible benefits**
 - Difficult to quantify, but may be more important, e.g.,
 - increased flexibility of operation
 - higher quality products/services
 - better customer relations
 - improved staff morale
- **How will the benefits accrue?**
 - When - over what timescale?
 - Where in the organization?

Analyzing Costs vs. Benefits

- Identify costs and benefits according to a checklist like on the previous slide and assign values to them.
- Determine the cash flow in two scenarios with/without a given set of requirements and compare the development over time, e.g. 3-5 years.
 - Calculate Net Present Value for all future costs/benefits, considering aspects like inflation and interest.

- Calculate the Return on Investment:

$$\text{ROI} = \frac{\text{Lifetime benefits} - \text{Lifetime costs}}{\text{Lifetime costs}}$$

- Calculate the break-even point, i.e. the number of years it takes to pay back the accrued costs:
 - $\text{Accrued Cost (initial + incremental)} < \text{Accrued Benefit}$

Exhaustive Comparison

- **A prioritization is a (partial) order on the elements to be prioritized.**
 - It can be established by comparing all pairs of requirements – at the cost of $O(n^2)$, where n is the number of requirements.
 - For small n , this can be done (e.g. XP “card game”).
- **For large n , segmentation helps to some degree.**
 - Roughly classify all requirements into m subclasses S_i ($m \in \{3, \dots, 5\}$ is often useful), plus a bucket B that cannot be classified into one of the S_i .
 - Apply this procedure recursively to each subclass S_i , until a given size of the subclasses is reached.
- **However, this approach assumes transitivity of priority, which may not be the case in some peoples view.**
- **Also, it only establishes relative priority, which does not help in finding an absolute cut-off.**

Analytic Hierarchy Process

- These problems can be addressed by the analytic process hierarchy (AHP) approach.
- Assume there is a set of n requirements. Arrange them in a $n \times n$ matrix and compare each pair of requirements.
 - For element (x,y) in the matrix enter:
 - 1 - if x and y are of equal priority, e.g. $x=y$
 - 3 - if x is slightly more preferred than y
 - 5 - if x is strongly more preferred than y
 - 7 - if x is very strongly more preferred than y
 - 9 - if x is extremely more preferred than y
 - ...and for (y,x) enter the reciprocal.
- Estimate the eigenvalues by averaging over normalized columns.
 - Normalize the columns (j): $r_{i,j} := r_{i,j} / \sum_{k=1}^n r_{k,j}$.
 - Normalize the rows (i): $r_{i,j} := r_{i,j} / \sum_{k=1}^n r_{i,k}$.
- This gives a value for each requirement based on estimated percentage of total value of the project.

AHP example

| | | | | | | | | | | | sum / #rows | | |
|----------------|----------------|----------------|----------------|----------------|---|----------------|----------------|----------------|----------------|----------------|--|------|------|
| | R ₁ | R ₂ | R ₃ | R ₄ | Normalize columns  | | R ₁ | R ₂ | R ₃ | R ₄ | Sum rows  | | |
| R ₁ | 1 | 1/3 | 2 | 4 | | R ₁ | 0.21 | 0.18 | 0.18 | 0.48 | | 1.05 | 0.26 |
| R ₂ | 3 | 1 | 5 | 3 | | R ₂ | 0.63 | 0.54 | 0.45 | 0.36 | | 1.98 | 0.50 |
| R ₃ | 1/2 | 1/5 | 1 | 1/3 | | R ₃ | 0.11 | 0.11 | 0.09 | 0.04 | | 0.34 | 0.09 |
| R ₄ | 1/4 | 1/3 | 3 | 1 | | R ₄ | 0.05 | 0.18 | 0.27 | 0.12 | | 0.62 | 0.16 |

- This procedure yields a percentage value of the expected contribution of each requirement.
- Sorting by contribution yields an absolute priority.
- Also, cut-off can be established as an absolute value.
 - For instance, a cut-off of 10% will exclude R₃.

Conflicts and Dependencies

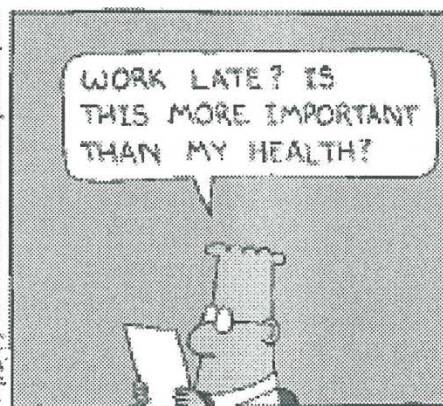
- **Another obstacle in the requirements arbitrage process is the fact that requirements are not isolated – there are typically rich relationships between them.**
- **Dependencies**
 - Requirements might depend on each other, e.g. R1 depends on R2.
 - If R2 has low priority (e.g., low ROI, risk, ...), but R1 has high priority, R2 might have to be implemented anyway.
 - If the priority is based on a ROI calculation, we might compute
$$\mathbf{ROI}(\{R_1, R_2\}) = \mathbf{ROI}(R_1) + \mathbf{ROI}(R_2).$$
 - If the priority is based on risk assessment, however, this might not hold.
- **Conflicts**
 - Requirements might also be mutually exclusive. If there are two such requirements that both have high priorities, we cannot avoid a confrontational decision. Luckily, this is rare.



EMAIL: SCOTTADAMS@AOL.COM



© 1997 United Feature Syndicate, Inc.



WWW.UNITEDMEDIA.COM





Prof. Dr. Harald Störrle
Danmarks Tekniske Universitet (DTU)

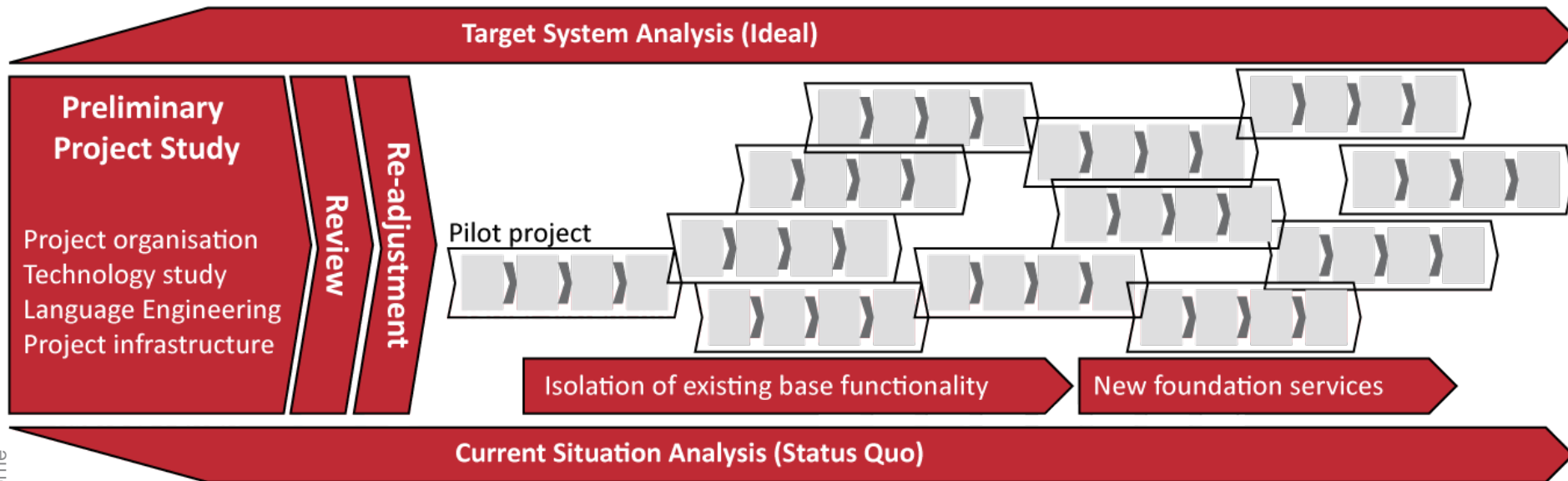
Chapter 12.3

Project Control with Requirements

DTU course 02264

Requirements in Project Management

- If a decision has been taken to implement a given set of requirements, some kind of project management ensues.
 - Using the requirements to plan the project and monitor its progress is a very good idea.



- Requirements ("Feature Sets") can also be used for planning ahead, e.g., in the definition of product families.

Requirements in Maintenance

- Once the initial increment of a system has entered integration, there will be change requests, e.g. for correcting errors, and for adding enhancements, improvements, and new features.
 - This is effectively the start of the maintenance phase.
- Since dealing with change requests is very similar to dealing with original requirements, it is a good idea to use the same process and tools for the two.
- However, there will be changes to some of the requirements' attributes.

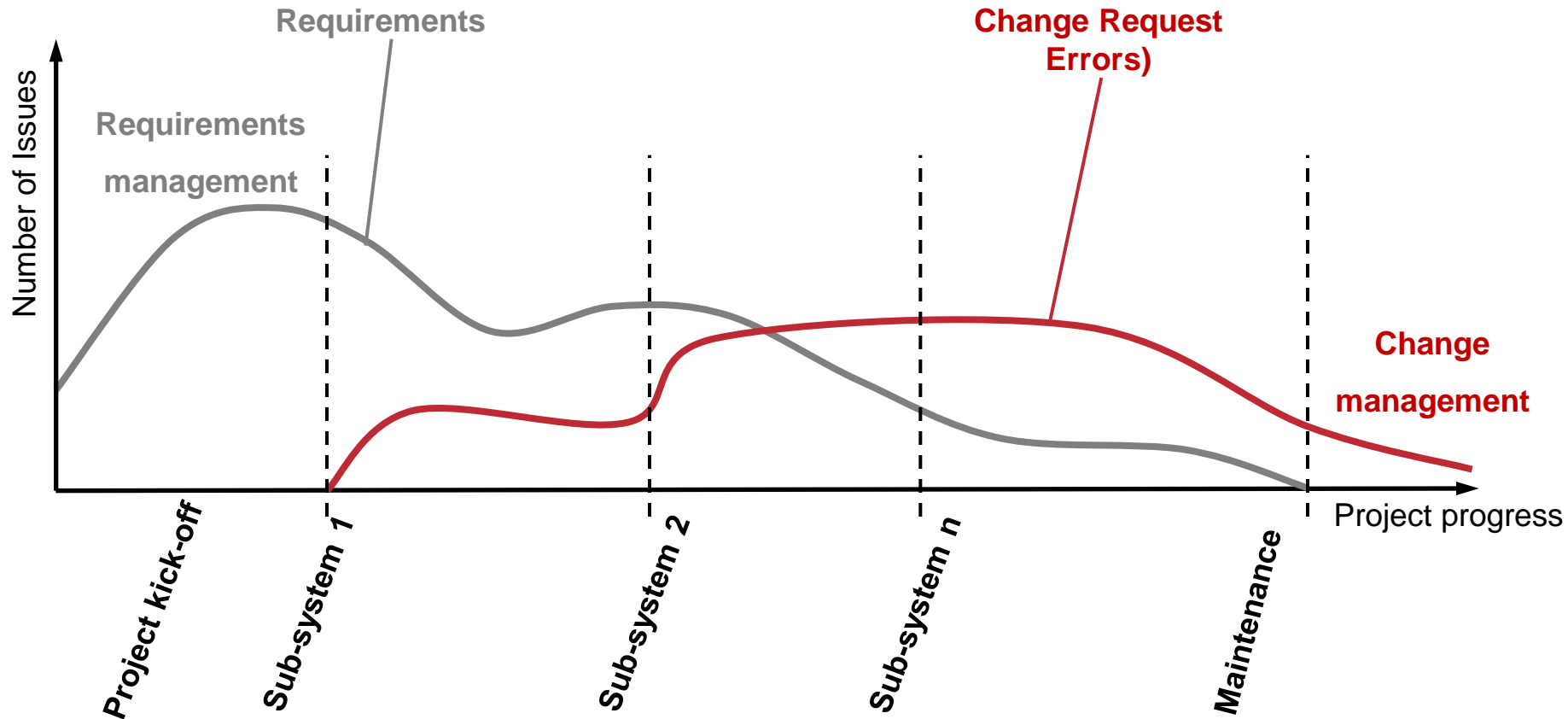
Kind: distinguish between original requirements and change requests

Source: must allow reference to an issue tracking system

Status: must allow for new issue states (e.g. “resolved” or “pending”)

Work Packages: must allow reference to artifacts or steps in the release plan

Requirements vs. Changes over time



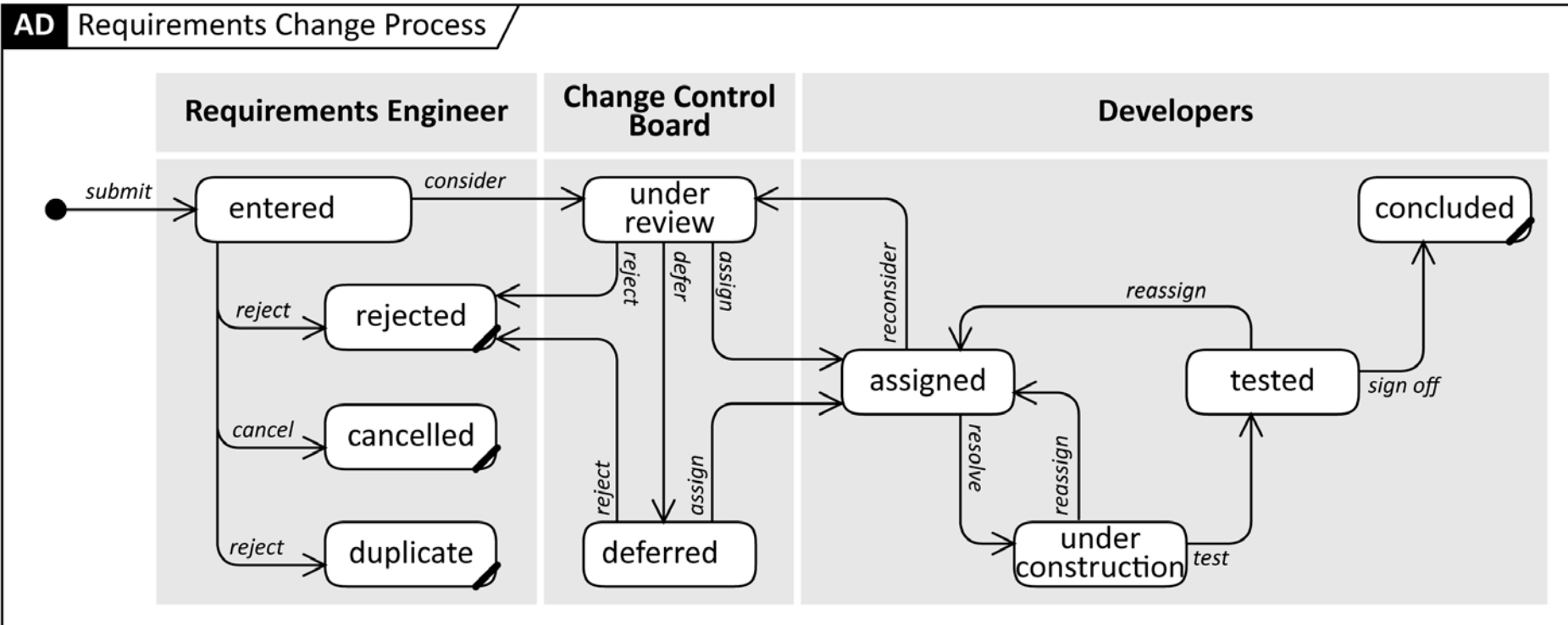
Errors vs. Changes

- **Surveys on large numbers of projects report requirement change rates of 3-10% per year, or 25% over the project lifetime.** [Jones 1994]
- **In long running projects, asymptotically, all the effort goes into maintenance, which includes the maintenance of requirements.**
 - While some parts of the system (and the requirements) stay virtually unchanged over the years, at the time of decommissioning, most of the system has been added or changed as compared to the initial release.
- **Requirements are the interface between client and supplier. Thus, they often have contractual character. This can be consequential.**
 - If a defined requirement is not implemented, the client is entitled to a free improvement of the delivered piece of work.
 - If a the client demands a change to a defined requirement, on the other hand, the provider is entitled to charge for the extra work.

Tracing

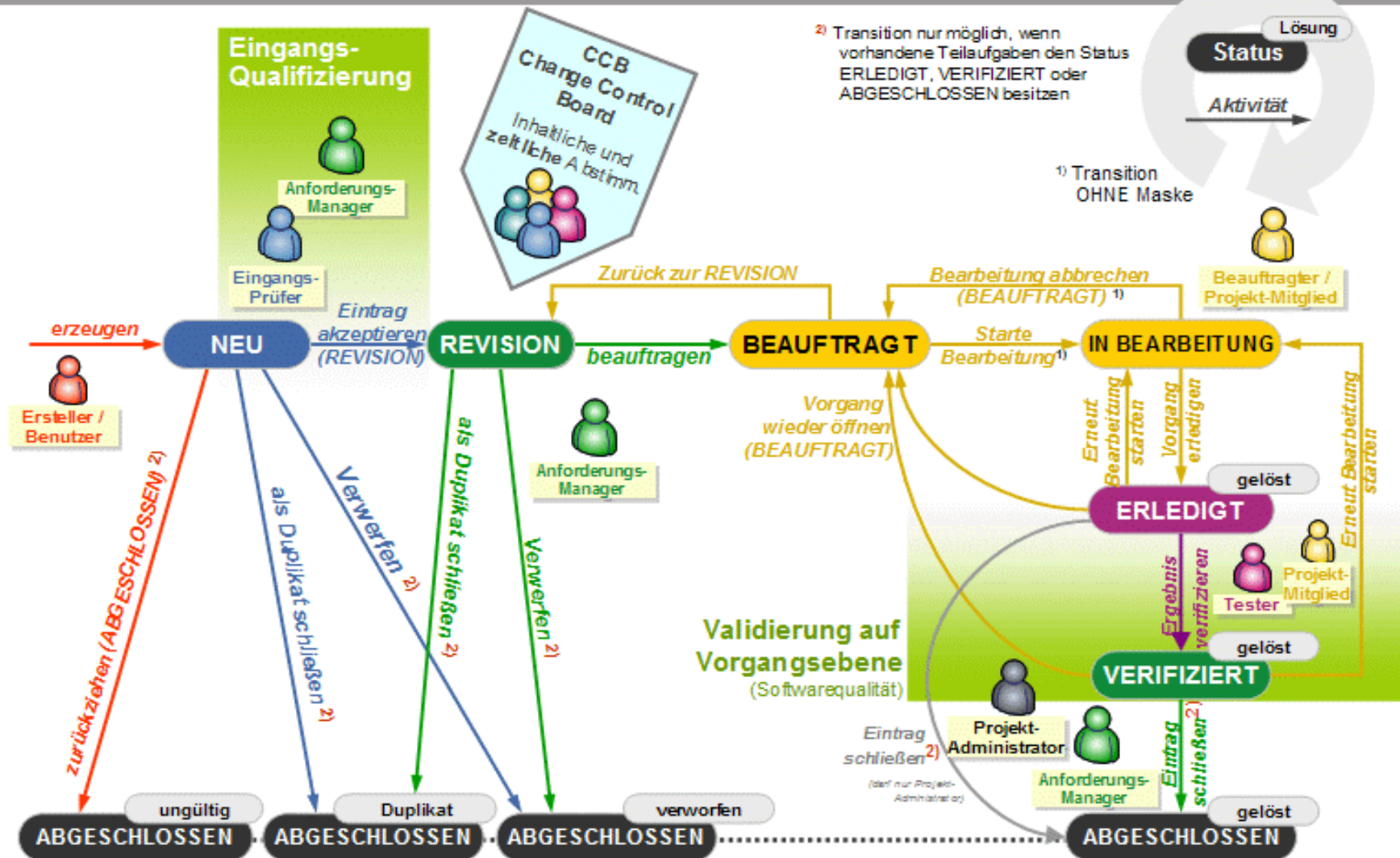
- **Consider the following integrity RQA.**
 - R1: “There may be no unauthorized access to the lending records.”
- **In order to make this requirement verifiable, one might want to replace it by statements such as these:**
 - R1.1: „Accessing the lending records must be secured by a login procedure involving a personal secret password no shorter than 8 symbols.
 - R1.2: „The lending records must be stored encrypted with AES-128.“
- **However, now a required quality attribute has been turned into two functional requirements, that raise a couple of questions.**
 - Do they actually satisfy the original requirement?
 - If so, is this the best way to do it? Is this the most cost-efficient way?
- **We will not be able to answer these questions unless we keep the original requirement, too, and record the relationship to the two new requirements.**
 - This is of paramount importance in all domains where certification plays a big role (cf. DO-178A/B/C, the Ariane example).

A Real-Life Requirements Lifecycle

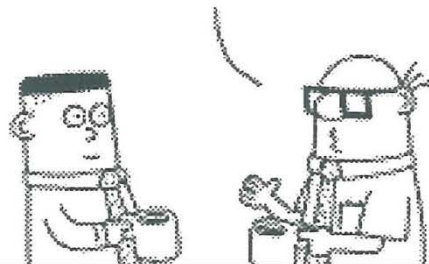


Lebenszyklus von Anforderungen

Workflow – Anforderung

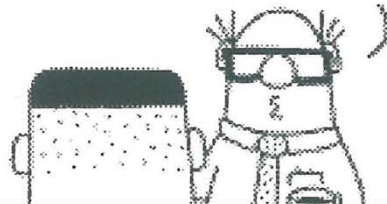


THE FIRST WEEK
AFTER GETTING AN
ASSIGNMENT IS
CALLED "THE WALLY
PERIOD."



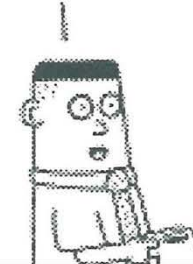
www.dilbert.com scottadams@aol.com

NEVER DO WORK
DURING THE WALLY
PERIOD BECAUSE
MOST TASKS BECOME
UNNECESSARY WITH-
IN SEVEN DAYS.



12/10/02 © 2002 United Feature Syndicate, Inc.

I WANT
A PERIOD
NAMED
AFTER
ME!



WHOA,
ASOK. THAT
TAKES
MANY
YEARS OF
NON-WORK.





Prof. Dr. Harald Störrle
Danmarks Tekniske Universitet (DTU)

Chapter 12.4

Requirements Management Tools

DTU course 02264

Tools for Requirements Management

- **When the set of requirements is very small and the project duration is short, there are only very few requirements management activities, so that tool support is unnecessary.**
- **However, already rather modest project sizes benefit from tool support.**
 - Consider an MSc-thesis about some kind of implementation.
 - There might be around 20 requirements to be tackled in 5 months.
 - Keeping track of these in your head alone is very hard and error prone.
- **The most basic tool is a pen-and-paper table or a spreadsheet.**
 - Use one row per requirement and columns for various aspects.
- **More advanced tools will allow you to link requirements for tracing, support versioning, and reporting.**
- **Unfortunately, there are not many specialized tools, they are not very good, and most of them are simultaneously very expensive.**

Requirements Attributes (Elicitation)

| Stage | Attribute | Description | Values |
|-------|-----------|-------------|--------|
|-------|-----------|-------------|--------|

Elicitation

| | | |
|--------------|---|---|
| Identifier | unique and persistent identifier | project specific, e.g. integers |
| Name | descriptive term, possibly phrase | proper name, phrase |
| Description | brief text describing what is included in this requirement | 3-10 sentences, at most two paragraphs |
| Rationale | A justification of this requirement: why is it being selected | 1-3 short sentences OR reference to a goal |
| Source | reference to origin of requirement | reference to documents, workshops, individuals, existing systems etc. |
| Contributors | the person who originally wrote this requirement, and any other persons who changed it afterwards | reference to the project organisation chart |
| Details | A more detailed treatment of this requirement | reference to an external document |
| Remarks | any additional remark, e.g. comments or open questions | text in project language |

Requirements Attributes (Elaboration)

| Stage | Attribute | Description | Values |
|-------|-----------|-------------|--------|
|-------|-----------|-------------|--------|

Elaboration

Type

classification of requirement

- feature ("functional req.")
- quality ("non-functional req.")

Level

granularity level or scope of the requirement

- market / domain
- product
- feature
- component

Derived From

(1) Reference to a requirement that has been split up into several smaller requirements, that collectively replace the original requirement.
(2) Reference to a requirement

reference to an obsolete requirement

Acceptance Test

(1) operational procedure to test this requirement
(2) quantification of minimum acceptable quality
(3) reference to another artifact detailing the acceptance criteria such as a test class or test specification document

(1, 2) text in project language
(3) path on project drive, CM system etc.

Requirements Attributes (Management)

| Stage | Attribute | Description | Values |
|-------|-----------|-------------|--------|
|-------|-----------|-------------|--------|

Management

Priority

the priority of this requirement relative to the other requirements in this list as seen by the client

project specific, usually a small set of priority classes, e.g. A..C, or 1..10
does not necessarily imply sequence of resolution of requirements

Prerequisites

reference to other requirements from this list that need to be satisfied before this one can be treated

requirement identifier list

Conflicts

reference to other requirements from this list that may be in conflict (only one of them can be satisfied at once)

requirement identifier list

Responsible

Person responsible for managing this requirement

Name, Position

Requirements Attributes (Transition)

| Stage | Attribute | Description | Values |
|------------|--------------|--|--|
| Transition | Predecessor | reference to another requirement from this list that described a previous, now obsolete version of this requirement | requirement identifier |
| | Is Part Of | reference to another requirement from this list that includes this requirement | requirement identifier |
| | Work Package | reference to the work package in which this requirement is being addressed as seen from development organization (acknowledged by client); ideally a task number from an issue tracker/CM system | reference to a work package in which this requirement is addressed (independent from priority) |
| | Kind | kind of requirement | project specific, e.g.: - bug - improvement - addition |

Sequence of elaboration

- **Not all the requirements attributes are needed at once.**
 - This is only necessary for: ID, name, Description, Rationale, Justification.
- **Some can be filled automatically by a tool.**
 - This applies to: (initial) status, Source, some changes
- **The attribute “Level” must be filled before setting “Priority” because it makes only sense to determine priorities of things at the same level.**
- **Acceptance test should be created only after setting the priority in order to focus the effort to where it is absolutely needed.**
- **Relationships are usually defined last, because they are management tools rather than elicitation tools, and they can only be established if the related elements are in place already.**
 - One possible exception is “Part of” which is often determined together with “Level”, but since that may be established only after all requirements are known, this aspect has to be revisited at the end.
- **“Responsible” is determined last since responsibilities should be known in full before assigning them/taking them on.**
 - Also, many people tend to stop thinking about this requirement as soon as somebody else’s name appears in this field.

Project Customization

- **Most projects will want to adapt and modify the requirements attributes in one way or another.**
 - Usually the motivation is to achieve a better fit between the project organization and the requirements attributes.
 - Another common reason is to allow the usage of existing/specific tools that are hard to customize.
- **Here are the most commonly customized attributes and alternative values/properties for them.**

Identifier: Anything from Integers via highly structured compounds to arbitrary strings

State: Requirement States and Transitions, with Roles/Rights

Sources: Acceptable Requirements sources

Level: various enumerations

Type: various enumerations

Acceptance test: might support/restrict values

Priority: enumeration of values, possibly computed by other factors

Predecessor: restriction to used Identifiers, check that predecessor is in state obsolete, properties of relationship (asymmetry, transitivity)

Part Of: restriction to used Identifiers, check that Container is of same or larger Level, , properties of relationship (asymmetry, transitivity)

Prerequisites: restriction to used Identifiers, check that relationship is asymmetric, warn if conflict among prerequisites, properties of relationship (transitivity)

Conflict: restriction to used Identifiers, check that relationship is symmetric, , properties of relationship (transitivity)

Work Package: Access to CM system or similar

Responsible: Access to list of names/positions in project organization chart

Existing RE Tools

- **There are some „true“ RE tools that have been created specifically for this purpose.**
 - They can be considered „heavy duty“ and are useable in the largest projects.
 - These tools are often very expensive (5-figures, €), require a considerable learning effort, and are not easy to maintain and operate.
 - They do offer all kinds of integrations and reporting facilities, but less in the department of advanced functionalities.
- **Here are some examples of commercial tools.**
 - Telelogic (now IBM): DOORS; Rational (now IBM): RequisitePro; Borland: CaliberRM; Compuware: Reconcile; NCH: Miro.BAS; Polarion: Polarion; TCP / QA Systems: IRqA; Serena: RTM Workshop; NoMagic: MagicRQ
- **There are only few examples of free RE tools (e.g., OSRMT).**
- **Alternatively, Bug/Issue tracking systems are sometimes used for the management part (e.g., Jira, Bugzilla).**

JIRA:

Using a Bug Tracker for RE

Create Issue - mgm JIRA - Windows Internet Explorer

https://jira.mgm-tp.com/jira/secure/CreateIssue.jspa

Google

Datei Bearbeiten Ansicht Favoriten Extras 2

Google Los geht's! Vuze Suche im Internet Ask What's Hot What's New Featured News Weather

Create Issue - mgm JIRA

mgm technology partners

User: Wolfgang Glock Filters Profile Log Out

HOME BROWSE PROJECT FIND ISSUES CREATE NEW ISSUE ADMINISTRATION QUICK SEARCH:

Create Issue

Step 2 of 2: Enter the details of the issue...

Project: Demo Anforderungsmanagement

Issue Type: Requirement

Field Tab Workflow-Info

* Summary:

* Anforderungsart: funktional
Daten / Information
Look and Feel
Benutzbarkeit
Performance / Zuverlässigkeit

Die Art der Anforderung
Mehrfachauswahl durch Halten von "Strg" + Click

Priority: Normal

Assignee: - Automatic - Assign to me

* Reporter: wglock
Start typing to get a list of possible matches.

Component/s: Unknown
GIS
SAP

Description:

Die Beschreibung sollte folgende Informationen so detailliert wie möglich enthalten:

Fertig

Internet 100%

JIRA:

Using a Bug-Tracker for RE

Create Issue - mgm JIRA - Windows Internet Explorer

https://jira.mgm-tp.com/jira/secure/CreateIssue.jspa

Google

Datei Bearbeiten Ansicht Favoriten Extras 2

Google Los geht's! Suche im Internet Ask What's Hot What's New Featured News Weather

Create Issue - mgm JIRA

mgm technology partners

User: Wolfgang Glock Filters Profile Log Out

HOME BROWSE PROJECT FIND ISSUES CREATE NEW ISSUE ADMINISTRATION QUICK SEARCH:

Create Issue

Step 2 of 2: Enter the details of the issue...

Project: Demo Anforderungsmanagement

Issue Type: Requirement

Field Tab Workflow-Info

* Summary:

* Anforderungsart: funktional
Daten / Information
Look and Feel
Benutzbarkeit
Performance / Zuverlässigkeit

Die Art der Anforderung
Mehrfachauswahl durch Halten von "Strg" + Click

Priority: Normal

Assignee: - Automatic - Assign to me

* Reporter: wglock
Start typing to get a list of possible matches.

Component/s: Unknown
GIS
SAP

Description:

Die Beschreibung sollte folgende Informationen so detailliert wie möglich enthalten:

Fertig

JIRA:

Entering a single requirement

Neuer Vorgang - ELSTER - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:8080/secure/CreateIssue.jspa

teamWARP Login Page IEEE - the world's le... LEO Deutsch-Englisc... Strato - SSL Login Wolfgang Test JIRA LocalWiki SourceForge.net

Benutzer: Wolfgang Glöck Verlaufs | Filter | Profil | Abmelden

STARTSEITE PROJEKTÜBERSICHT VORGÄNGE SUCHE **NEUER VORGANG** ADMINISTRATION

SCHNELLSUCHE:

Neuer Vorgang

Schritt 2 von 2: Vorgangsdetails

Projekt: Testprojekt

Vorgangstyp: Fehler

* Zusammenfassung:

Priorität: Bedeutend

Datum fällig:

Komponenten: Unbekannt

betrifft Version(en): Unbekannt

Umsetzung geplant in Version(en): Unbekannt

Zuständigkeit: Für die Erfüllung dieses Eintrages notwendige Zulieferer oder Entwicklungsgruppen

* Zuweisen an: - Automatisch - mir zuweisen

* Ersteller: wglock

Umgebung:

Nennen Sie die komplette Betriebssystem-Version (z.B. Solaris-9; Windows XP Home). Die Auflistung mehrere BS ist ebenfalls möglich. Bitte geben Sie auch installierte Service-Packs mit an. Diese Information erhalten Sie auch mit dem Konsolenbefehl 'ver' unter Windows/Linux.

Browser-Version:

Bitte nennen Sie die komplette(n) Browser-Version(en), die betroffen sind. (z.B. Mozilla Firefox 1.0.7, MS Internet Explorer 6.0). Diese Information können Sie auch dem Konfigurationsassistenten des ElsterOnline-Portals entnehmen!

JAVA Laufzeitumgebung:

'Development Kit' oder 'Runtime' version: Fügen Sie hier die Ausgabe des Befehls 'java -version' ein. Falls Sie J2EE einsetzen, dann tragen Sie hier bitte auch die Ausgabe des Befehls 'j2ee -version' oder 'asadmin version' ein. Diese Information können Sie auch dem Konfigurationsassistenten des ElsterOnline-Portals entnehmen!

JAVA-Skript aktiviert?:

☐ None

☒ aktiviert

☐ deaktiviert

Haben Sie in Ihrer Browser-Konfiguration JAVA-Skript aktiviert? Diese Information können Sie auch dem Konfigurationsassistenten des ElsterOnline-Portals entnehmen!

Done

JIRA:

Displaying a single requirement

The screenshot displays the JIRA Testfehler interface in a Mozilla Firefox browser window. The address bar shows the URL `http://localhost:8080/browse/TSP-2`. The browser's bookmark bar includes links to teamWARP Login Page, IEEE, LEO Deutsch-Englisc..., Strato - SSL Login, Wolfgang Test JIRA, LocalWiki, SourceForge.net, and Dashboard - MGM-Wiki. The JIRA interface has a green header bar with the user name "Benutzer: Wolfgang Glock" and links for "Verlauf", "Filter", "Profil", and "Abmelden". Below the header is a navigation bar with tabs: "STARTSEITE", "PROJEKTÜBERSICHT", "VORGÄNGE SUCHE", "NEUER VORGANG", and "ADMINISTRATION". A search bar labeled "SCHNELLSUCHE:" is on the right. The main content area is titled "Vorgangsdetails" and shows the following information:

- ID:** TSP-2
- Typ:** Fehler
- Status:** Offen
- Priorität:** Geringfügig
- Bearbeiter:** Wolfgang Glock
- Ersteller:** Wolfgang Glock
- Beobachter:** 0 (Ansicht)

Below this information is a section for "Workflow-Aktionen" with several actions: "Starte Bearbeitung", "Thema gelöst", "Eintrag schließen", and "Aktionen". The "Aktionen" section lists various actions like "Zuweisen", "Duplizieren", "Kommentar", "Erstellen", "Löschen", "Bearbeiten", "Verknüpfen", "Verschieben", and "Beobachten".

The main details section for "Testfehler" includes the following information:

- Erstellt:** 02. Feb 2006, 12:39 Uhr
- Aktualisiert:** 02. Feb 2006, 12:39 Uhr
- Fällig:** 03.03.2006
- Komponenten:** Keine
- betrifft Version(en):** Keine
- Umsetzung geplant in Version(en):** Keine
- Umgebung:** Windows XP
- Browser-Version:** Mozilla
- JAVA Laufzeitumgebung:** 1.5
- JAVA-Skript aktiviert?:** aktiviert
- Schritte zur Reproduktion des Fehlers:** Verwendete URL: <bitte geben Sie hier die von Ihnen verwendete URL an> Schritte:
- Erwartete Ergebnisse:** Passt schon
- Aktuelle Ergebnisse:** Passt auch
- Fehler-Meldung(en):** Keine
- Beteiligte:** Wolfgang Glock
- Zeit seit letztem Kommentar:** 4 Wochen, 6 Tage ago

The "Beschreibung" section contains the text "Dies ist ein Testfehler". Below this is a section for "Alle Kommentare Änderungsprotokoll" with a "Sortierreihenfolge:" dropdown. The text "Bis jetzt wurde dieser Vorgang noch nicht kommentiert." is displayed.

At the bottom of the page, there is a red banner with the text "EVALUATION LICENSE - Are you enjoying JIRA? Please consider purchasing it today."

JIRA:

Reporting

Navigator - ELSTER - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:8080/secure/IssueNavigator.jspa?reset=true&mode=hide&pid=12310180

teamWARP Login Page IEEE - the world's le... LEO Deutsch-Englisc... Strato - SSL Login Wolfgang Test JIRA LocalWiki SourceForge.net Dashboard - MGM-Wiki

Benutzer: Wolfgang Glock Verlauf Filter Profil Abmelden

STARTSEITE PROJEKTÜBERSICHT **VORGÄNGE SUCHEN** NEUER VORGANG ADMINISTRATION

Filter: Ansicht Bearbeiten Neu Verwalten

Sie benutzen zur Zeit eine neue, ungespeicherte Suche.
☐ Speichern als Filter

Zusammenfassung

☐ Projekt: Testprojekt
☐ Sortiert nach: ID absteigend

Aktionen

Navigator

Vorgänge 1 bis 2 von 2 Vorgängen insgesamt

Ansicht wählen:
Browser | Druckansicht | XML | Gesamter Inhalt | Excel (Standardansicht | Aktuelle Felder)

☐ Massenänderung: alle 2 Vorgänge
☐ Navigator konfigurieren

| Art | ID | Zusammenfassung | Bearbeiter | Ersteller | Prio | Status | Lösung | erstellt | aktualisiert | fällig |
|-----|-------|---|------------------|----------------|------|--------|------------|------------|--------------|------------|
| | TSP-2 | Testfehler | Wolfgang Glock | Wolfgang Glock | ↓ | Offen | UNERLEDIGT | 02.02.2006 | 02.02.2006 | 03.03.2006 |
| | TSP-1 | Dies ist eine exemplarische Anforderung | Nicht zugewiesen | Wolfgang Glock | ↑ | NEU | UNERLEDIGT | 30.01.2006 | 02.02.2006 | 01.01.2007 |

EVALUATION LICENSE - Are you enjoying JIRA? Please consider [purchasing it](#) today.

Powered by Atlassian JIRA™ the Professional Issue Tracker. (Enterprise Edition, Version: 3.4.3-#109) - [Bug/feature request](#) - [Contact Administrators](#)

Done

RSS



Prof. Dr. Harald Störrle
Danmarks Tekniske Universitet (DTU)

Chapter 12.5:

Tailoring the Requirements Toolbox

DTU course 02264

The necessity of Tailoring

- **There are many different techniques for specifying and managing requirements – which ones should we use?**
 - One size does not fit all.
 - Each method has their specific profile of strengths and weaknesses.
 - For many techniques, we do not have adequate evidence to assert usage conditions: common sense and experience will have to do.
- **Using an inappropriate technique might be worse than using no technique at all for several reasons.**
 - Disagreement about the approach can be distracting („method wars“) and disrupt the team's group dynamics.
 - Using methods typically comes with increased effort and/or cost.
 - The techniques may lead to properties of the system document that may not just be wasteful, but actually negative.
- **Imagine a scenario where using User Stories is demanded, while some team members prefer Use Cases.**
 - Convincing and training them requires effort and time.
 - Focusing on features may lead to neglecting qualities.

The Toolbox

- It is recommended practice to select and fix a set of techniques for a project, based on an initial estimate of the project's needs.
- We call this the „project toolbox“, and the process „tailoring“.
- During tailoring, a (brief) description of the „toolbox“ should be created.
 - The toolbox should be a project specific selection of existing proven practices, possibly with one or two additions of new “experimental” methods.
 - A justification of the decision must be provided.
 - The toolbox must be easily available, e.g., as a printed poster on the wall next to the coffee machine.
 - One team member should be appointed as responsible for maintaining the toolbox (the “tool smith”).
 - After the project, a post-mortem should be conducted to, among other things, assess the toolbox and the tailoring process.

Tailoring the Toolbox

- **Based on the toolbox description, and the project characteristics as known at the given point in time, techniques and tools will have to be selected.**
 - Techniques and tools are interdependent and have to be selected together.
 - Compromises are inevitable – and you will be criticized for them. So you better document the process and how all the relevant stakeholders pledged support for your approach.
 - Stakeholder mitigation techniques apply, throughout the process.
- **There is little or no scientific guidance on this process, even though it is extremely costly to get it wrong.**
 - Existing tailoring approaches (in particular VM'97), are perceived as heavy-weight and difficult, which only reflects and documents the appalling state of the practice in software engineering.

Example Toolboxes

| Project Type | Conventional Green Field Development | Administrative system main- tenance | Safety Critical Embedded | User Centered Design | Large Scale System |
|-------------------------|--|--|---|----------------------------|------------------------------------|
| Focus | End User | Productivity, Adaptation | Neighboring Systems | Stakeholder | Stakeholders Users, Operations |
| Software Process | Iterative plan- based | feature-driven ("agile") | Sequential plan, model-based supply chain | Prototyping | Cascading plan- based |
| Quality Assurance | Unit testing, Integration testing | Client validation | Verification Certification Rigorous Testing | User Validation | Testing, Review |
| Elicitation | Market research | user/client knowledge | Legal constraints | Field Study | Existing System, user knowledge |
| Specification Focus | Features | Features | Features, Qualities | Usage Patterns | Processes |
| Specification Format | Use Cases + scenarios | Bug reports, issue tracker entries, emails | Text, Models | Personas + scenarios | Process models, text |



Prof. Dr. Harald Störrle

Software Engineering Section
Department of Informatics and Mathematical Modeling
Technical University of Denmark
Richard Petersens Plads
Building 322, Room 024
DK-2800 Kgs. Lyngby

hsto@imm.dtu.dk
www.imm.dtu.dk/~hsto

