



Softgoals and Quality Requirements

Arlind Bejta



AGENDA

A Risk-Based, Value-Oriented Approach to Quality Requirements

Non-functional requirements in software engineering

Chung
Nixon
Yu
Mylopoulos



A Risk-Based, Value-Oriented Approach to Quality Requirements



Usually when requirements are elicited from stakeholders, they are often informally and qualitatively stated. As a result the developers build a system that doesn't meet the stakeholder's needs (delivers less or more than actually needed) and there can be a disagreement between parties about a given quality requirement.

To solve these problems we need to quantify all quality requirements.

QUANTIFICATION

“The response time must be fast”

Quantification means defining metrics to make requirements measurable.

quality requirement as that can be stated better as ...

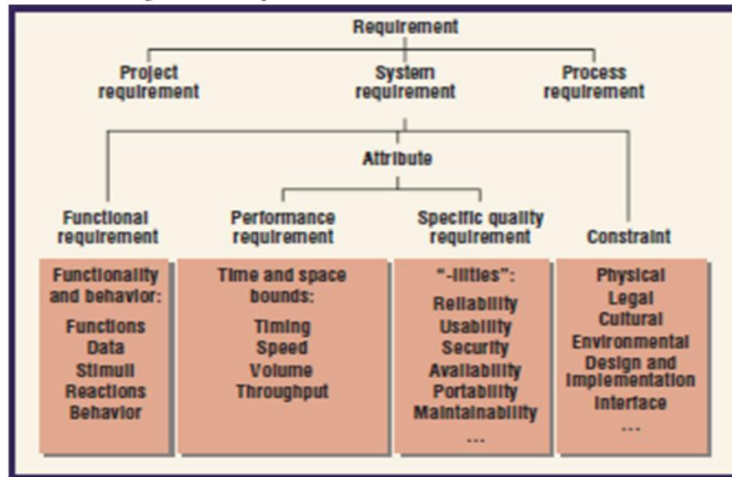
→

QUANTIFICATION

“The response time shall be less than 0.5 seconds in 98 percent of all user input action”

Goal is to motivate stakeholders to be as precise as possible to specify concrete threshold values.

Defining Quality Requirements



the term quality requirement covers those requirements that belong as system attributes, such as performance attributes or specific qualities.



Non-functional requirements in software engineering



Throughout the system development lifecycle, most approaches to system design are driven by functional requirements and intend to ignore non-functional requirements. This can lead to negative effects resulting in loss of reputation or potential customers.

→

The NFR Framework



<https://www.halogensoftware.com/upgrade/learn/how-to-improve-performance-manage-multiple-tasks-that-are-often-overlooked/improve-performance-manage-multiple-tasks-that-are-often-overlooked.html>



https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSUBxz_j9BWHR_oqZHEQV9HDOYais1RQH18HxQJARQ322HdFU0GM/w

the main idea behind this framework is to put non-functional requirements in the developer's mind

-In contrast to functionality-driven approaches, the NFR Framework uses nonfunctional requirements such as **security**, **accuracy**, **performance** and **cost** to drive the overall design process.

The NFR Framework-Design Process

Acquiring or accessing knowledge

Identifying nonfunctional requirements for the domain

Decomposing NFRs

Identifying "operationalizations"

let's explain the processes of NFR Framework:

acquiring or accessing knowledge about:

- the particular domain and the system which is being developed,
- functional requirements for the particular system, and
- particular kinds of NFRs, and associated development techniques

identifying "operationalizations" (possible design alternatives for meeting NFRs in the target system),

Decomposing NFRs

Identifying "operationalizations"

→

The NFR Framework-Design Process 2

Deal with ambiguities, tradeoffs and priorities, and interdependencies

Selecting operationalizations

Supporting decisions with design rationale

Evaluating the impact of decisions

dealing with: ambiguities tradeoffs and priorities, and interdependencies among among NFRs and operationalizations

Selecting operationalizations

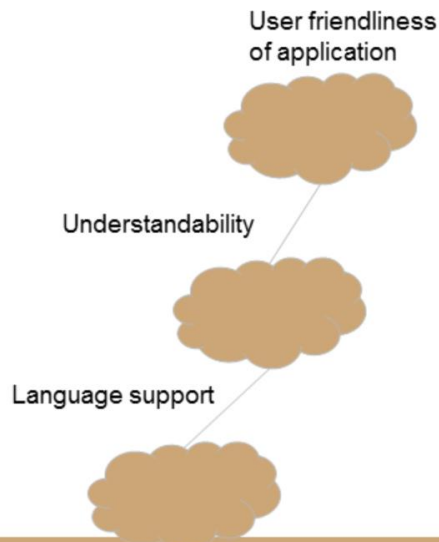
Supporting decisions with design rationale

Evaluating the impact of decisions

this is not a sequential approach or top-bottom approach. The development process can move up and down rather than following a strict top-down approach

→ Additionally, this Framework offers a structure for representing and recording the design in graphs, called softgoal interdependency graphs (SIGs).

Softgoal Interdependency Graphs



Major concepts of the Framework appear in the graphical form in SIGs.

- Softgoals, are shown as clouds.

- Main requirements are shown as softgoals at the top of a graph.

- Softgoals are connected by interdependency links, which are shown as lines,

- Interdependencies show refinements of "parent" softgoals downwards into other more specific, "offspring" softgoals. They also show the contribution (impact) of offspring softgoals upwards upon the meeting of other (parent) softgoals.

Catalogues

Knowledge about the particular types NFRs

Systematically organize development techniques

Implicit interdependencies among softgoals

Sources:

Textbooks, developer guides, handbooks

Specialized knowledge (Industry, Academia, Organization)

A very important aspect of the Framework is that developers can use their previous knowledge in catalogues.

There are 3 catalogues used:

1. Knowledge about the particular types of NFRs (such as security and performance)
2. The second catalogue is used to systematically organize development techniques to help meet requirements
3. Show implicit interdependencies (correlations, tradeoffs) among softgoals

The design knowledge in catalogues may come from many sources:

Textbooks, developer guides, handbooks

Specialized knowledge (Industry, Academia, Organization)

NFR FRAMEWORK- ACQUIRING DOMAIN KNOWLEDGE

Functional Requirements

Organization workload

Organizational priorities

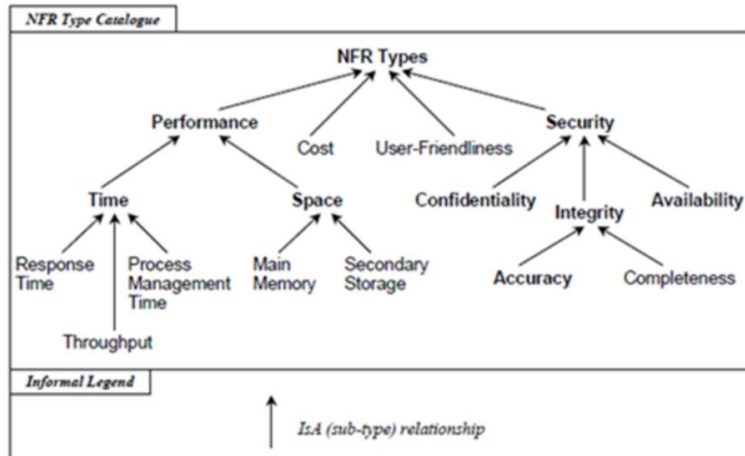
For example lets say that we have a system that manages credit cards.

Functional requirements include processes such as authorization of purchases, updating account etc.

Organizational workload includes the number of cardholders and merchants and expected daily volume of purchases.

Organizational priorities would be for instance cancellation of stolen cards etc.

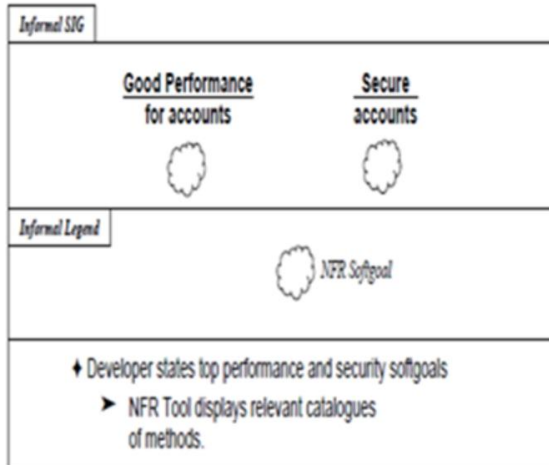
ACQUIRING AND CATALOGUING NFR KNOWLEDGE



https://files.ifi.uzh.ch/verghesingre-lims17/protected/ass/Chung_et_al2000_NFRAction.pdf

To provide a terminology and classification of NFR concepts, NFR type catalogues are used. The figure you see shows a catalogue of NFRs. As you can see, they are arranged in hierarchy ranging from more general (top) to more specific (bottom).

IDENTIFYING and DECOMPOSING NFRs



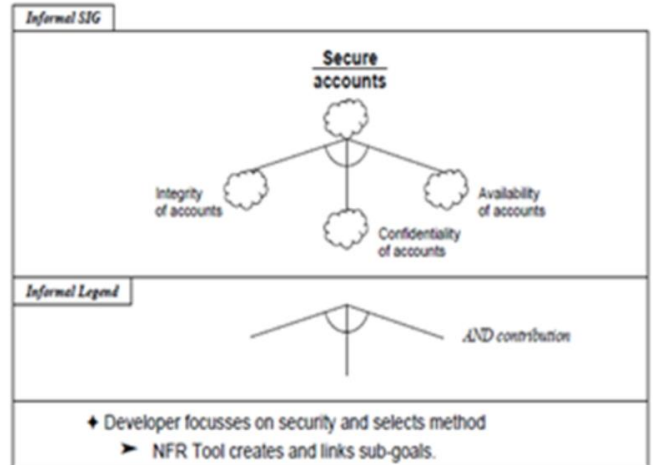
https://files.it-uzh.ch/research/teaching/re-ihw17/protected/ass5/C hung_er_al2000_nfrFwAction.pdf

When using the framework, first step is constructing an SIG by identifying the main non-functional requirement that a particular system should meet.

Example: Credit card system may include security of account information & good performance in storing and updating information. They're treated as softgoals to be achieved.

→

IDENTIFYING and DECOMPOSING NFRs



https://files.it-uzh.ch/research/ingre-ihw17/protected/aa5C/hung_et_al2000_nfrFwAction.pdf

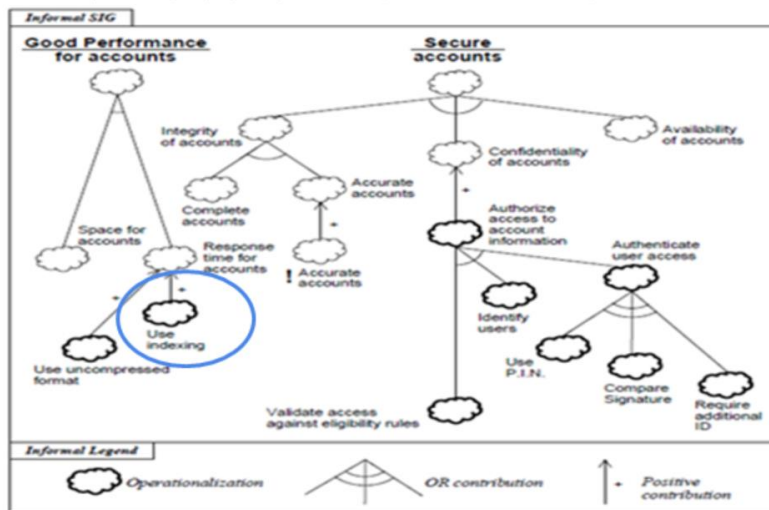
When decomposing a softgoal, the developer can choose to decompose its type or its topic.

Security softgoal can be broken down into smaller elements so that solutions can be found.

By treating high-level requirements as softgoals to be achieved, we can decompose it into more specific subgoals. As a result, the security softgoal can be further broken down into “Integrity of accounts,” “confidentiality, and availability

When all of several sub-softgoals are needed together to meet a higher softgoal, we say it is an AND type of contribution. In our example, all ofsprings (integrity, confidentiality and availability) are all met, then we say that as a group they will achieve the security softgoal

IDENTIFYING POSSIBLE OPERATIONALIZATIONS



https://files.ifi.uzh.ch/verg/teaching/re-lins17/protected/ass5/Chung_et_al2000_NFRAction.pdf

At some point, when the non functional requirements have been sufficiently refined, identifying possible development techniques to achieve the softgoals is possible.

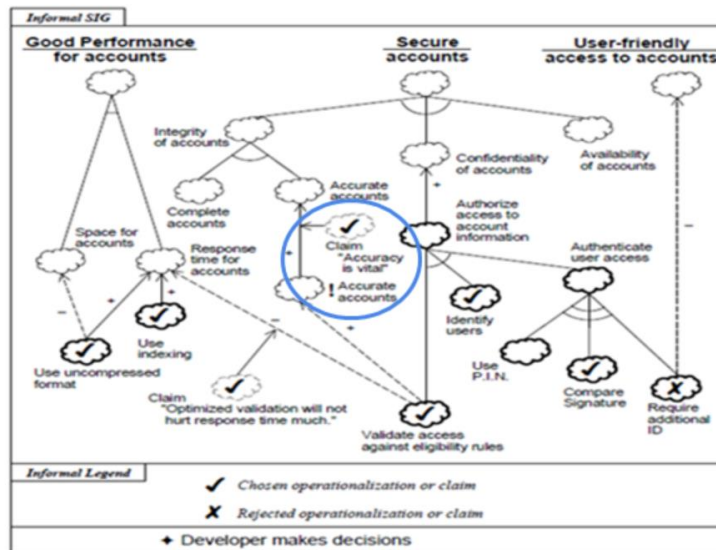
However, there is a gap between softgoals and development techniques. To close the gap, analysis is needed. Factors such as ambiguities, tradeoffs, priorities, have all to be taken into account.

Thus for example to minimize the response time we can combine a development technique (indexing) to satisfy a softgoal.

*OPERALIZATIONS are development techniques

*OR Contributions

RECORDING DESIGN RATIONALE & SELECTING AMONG ALTERNATIVES

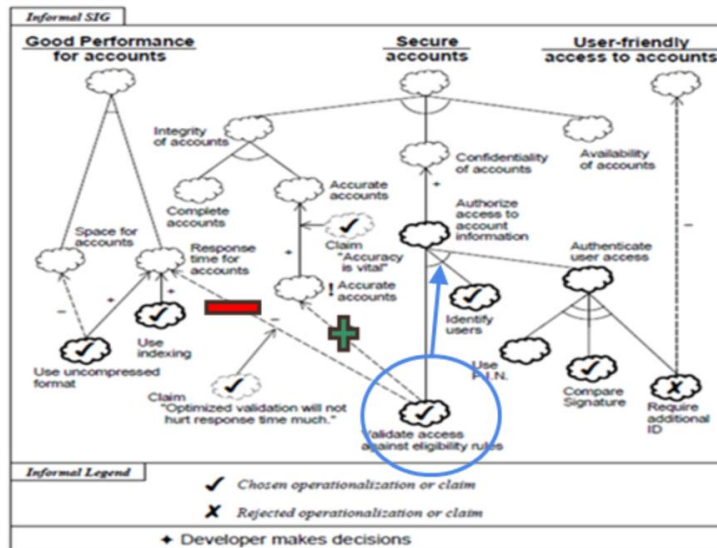


An important premise of the Framework is that design decisions should be supported by well-justified arguments or design rationale.

For example: stating a reason for prioritization. It is stated “Accuracy is vital”. This is called a **claim softgoal**

→

RECORDING DESIGN RATIONALE & SELECTING AMONG ALTERNATIVES



Here this tradeoff, "Validate access against eligibility rules" helps to achieve the more general operationalization, & has positive impact on the softgoal for accuracy of accounts, which is a priority.

However, it has a negative impact on the softgoal for good response time for accounts. In this case the developer weights the tradeoff and considers the tradeoff.

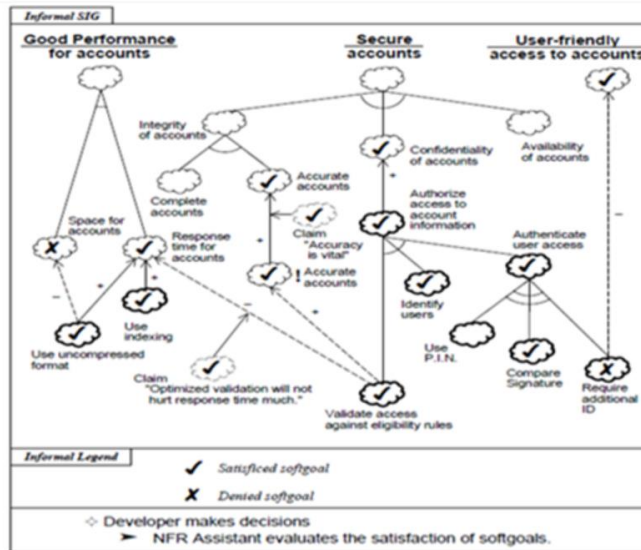
Selecting among alternatives

After considering different factors, now the developer can choose among the possible operationalizations in order to produce a target system.

Chosen operationalizations are marked by check mark and discarded operationalizations are marked with X. claims and tradeoffs are also either accepted or rejected.

A decision also can be left blank as not every operationalizing softgoal requires a decision.

EVALUATING THE IMPACT OF DECISIONS



https://files.ifi.uzh.ch/vergleichende-re-kinstl7/protected/ass/Chung_et_al2000_NFRAction.pdf

The evaluation of softgoals and interdependencies determines the impact of decisions. This indicates whether high-level softgoals are met. To determine the impact of decisions, both current and previous decisions are considered

To reflect the nature of design reasoning, the evaluation focuses on the question of whether a chosen option is “good enough” and meets the softgoal sufficiently. Since we deal with non-functional requirements, we can say that this style of reasoning is appropriate because meeting these requirements is often matter of degree and not a binary decision.



How much refinement is needed?





Anyone used such a framework?



Large systems and use of SIG!