

Guiding Architectural Decisions with the Influencing Factors Method

Pia Stoll
Industrial Software Systems
ABB Corporate Research
pia.stoll@se.abb.com

Anders Wall
Industrial Software Systems
ABB Corporate research
anders.wall@se.abb.com

Christer Norström
Computer Science and Electronics
Mälardalen University
christer.norstrom@mdh.se

Abstract

The Influencing Factors (IF) method guides the architect through stakeholders' concerns to architectural decisions in line with current business goals. The result is a set of requirements on software quality attributes and business goals and highlighted trade-offs among software quality attributes and among business goals. The IF method is suitable for sustainable software systems since it allows new concerns, resulting from changes in business goals, stakeholder concerns, technical environment and organization, to be added to existing concerns.

1. Introduction

For the architect it can be difficult or even impossible to satisfy all concerns from stakeholders in one architecture. Concerns are those interests which pertain to the system's development, its operation or any other aspects that are critical or otherwise important to one or more stakeholders. Concerns include system considerations such as performance, reliability, security and distribution [11].

Concerns are not always in line with stated business goals and do not always have positive impact on the product's market differentiator(s). For example the upper management wants to use standard hardware and software but this could have a negative impact on a product with high reliability as its market differentiator.

The sustainable software systems area is concerned with the architecting and designing of systems that are maintainable, supportable and modifiable during long life-times in the face of changing customer requirements and changing environments (hardware, commercial-of-the-shelf products, security threats, operating systems, communication standards). A sustainable system could have been developed according to plan, but still would not be able to cope successfully with change since the requirements for sustainable software systems will change over time.

Sooner or later the systems architect has to take architectural decisions in order to satisfy changes in both business goals and changes in functional and non-functional requirements. In order to be pro-active the architect should continuously analyze the changing concerns and not only wait for the concerns to be transformed into functional and non-functional requirements.

Requirements usually lack any trace of what concerns they originated from and therefore it is not clear what effect they have on business goals and quality attributes. This may lead to confusion among the developers implementing requirements they do not really understand the origin of. The analyzed concerns could contribute to a more complete requirement specification than the requirement list. In [12] Nancy G. Leveson proposes an approach to provide specifications that support human problem solving and where the outcome from the concern analysis could serve as input.

It would make the architects life easier and enable better communication among diverse stakeholders if there was a time-saving method for the analyzing of concerns and their influence on the architecture. The architect could then put forward the implications of each concern and their internal relations and help the stakeholders to make rational decisions on what concerns should be prioritized. Concerns may be very stakeholder-related and it can be beneficiary to filter the concerns so that only the facts of the concerns enter the analysis and not the relation to the stakeholders.

This paper presents the Influencing Factors, IF, method which targets this challenge. The method collects concerns, extracts influencing factors from the concerns and analyzes those for their influence on business goals and software quality attributes. The result is a business goal oriented prioritization of software quality attributes.

The *influencing factor* is a factor that affects architecture design [10]. The influencing factors are extracted from the stakeholders' concerns in the IF method. The global analysis described in [10] analyzes

each influencing factor's detailed impact on the design, the schedule and the components and not on specific business goals or software quality attributes.

From a line managers concern; "We want to implement the system in Java since our developers are skilled in and enjoy working with Java" the influencing factor "Implement the system in Java" is extracted. The same influencing factor can influence different business goals and software quality attributes. The same influencing factor as above; "Implement the system in Java" can also be extracted from an upper management concern; "We want to use Java since it is too costly to re-design the system in C#". In this concern the business goal is to reduce total cost of ownership and in the first concern the business goal is to maintain jobs of the workforce on the legacy system.

The method is illustrated with two industrial case studies. The first case study was performed on the upgrade of a large legacy system and the second case study on the re-factoring of an existing system. The two case study systems had a diverse set of stakeholders, such as software architect, system architect, developers, testers, product management, line management, engineers, users, and many more. Both systems suffered from an unclear understanding of what concerns were the most important. Should the architects propose architectures that try to solve all concerns, which in practice would be impossible, or should they focus on some of them?

The remainder of this paper is organized as follows; section 2 describes current research and the definition of business goals and software quality attributes used in the method, section 3 presents the relationships of enterprise, system and software architecture, section 4 presents the three steps of the method, section 5 and 6 present two case studies where the method was applied. Section 7 presents the conclusions of the work with the IF method and finally, future work is presented in section 8.

2. Business goals and software quality attributes

A software product interfaces with people who have an interest or share in the product's business or enterprise. These people are the stakeholders of the system. The stakeholders are users, developers, management, sales & marketing people, support engineers etc. The stakeholders experience advantages as well as disadvantages with the product depending on the concerns they find that the product should satisfy. Concerns can influence both business goals as well as system quality attributes. For example: In order to produce flexible, adaptable applications, the reflection

architectural pattern, which provides a mechanism for changing structure and behavior of software systems dynamically [13], can be used. This pattern increases the modifiability of the system. But software reflection techniques may put high requirements on software developers since this way of coding is more complex than normal static programming. The complexity can lead to longer development time and therefore affect the business goal "Reduce cost of development" or "Time to Market" in a negative way. Analyzing concerns for their influence on business goals and software quality attributes is therefore necessary in order to find a suitable architectural solution. To be able to analyze the concerns influence on the business goals the business goals can be categorized. Len Bass and Rick Kazmann have categorized business goals from a number of ATAM evaluations [1]. Bass' and Kazmann's five categories are: (1) "Reduce total cost of ownership", (2) "Improve capability/quality of system", (3) "Improve market position", (4) "Support improved business processes", and (5) "Improve confidence in and perception of the system". The category "Reduce total cost of ownership" includes the subcategory "Reduce cost of development" and in the category "Improve capability/quality of system" the subcategories "Performance" and "Ease of use" are found. The categories facilitate the analysis of business goal changes with time in the IF method.

As well as having its own set of business goals a company or domain usually have its own sets of system quality attributes. In the case studies where we have applied the IF method we have used the six system quality attributes discussed in [2]; (1) Modifiability, (2) Security, (3) Usability, (4) Performance, (5) Availability and (6) Testability. In [2] it is argued that reliability is a part of the availability quality and reliability tactics are a sub set of the availability tactics. In the two case studies where the IF method was applied availability has been used to cover both availability and reliability related aspects. However, some voices have argued that reliability would have been better than availability. The architecture team or architect applying the IF method should therefore clarify with the architecture's stakeholders what quality attributes are to be used in the beginning of the analysis.

To be able to analyze the concerns for a software product the concerns first must be extracted from the stakeholders. Interviews, document reading and personal experiences are some ways of extracting concerns. The Quality Attribute Workshop, QAW [3], is an established method to extract concerns in the form of scenarios from stakeholders in order to find prioritized business goals and software quality attributes. The QAW method can be used in

conjunction with the Attribute Driven Design method, ADD [6], to achieve an architecture where all important quality attributes are considered. The QAW lets stakeholders put forward their concerns in the form of scenarios in a round-robin fashion in a one day's workshop. In order to prioritize the scenarios the stakeholders vote. This method gathers a large variety of stakeholders and let them meet and hear each others concerns. In our second case study the QAW result is incorporated with the IF method to get a complete picture of the prioritized business goals and software quality attributes.

3. Enterprise, System and Software Architecture

The influencing factors are part of the stakeholder concerns and include trends, technical environment, previous experiences and market demands etc, Figure 1. The stakeholder concerns can have many influencing factors.

The stakeholder concerns change over time as the influencing factors change over time. New trends, experiences and technical environments influence business goals and system quality attributes. For every change in concerns the software architect faces new business goals to satisfy, and new software quality

attributes to achieve in the system respective the software architecture.

Business goals are manifested by the enterprise architecture which includes business processes and business structures, e.g. a company which sells a product needs a sales division and probably a marketing division. The enterprise architecture provides a basis for the system architecture, e.g. a company developing a safety critical software product needs a team of safety experts and processes for testing and verifying the safety properties of the product.

The business goal categories presented in Section 2 are strongly related to the enterprise architecture's business processes and business structures shown in Figure 1. The first category of "Reduce total cost of ownership" means reducing cost for the entire enterprise architecture. The second, third and fifth categories; "Improve capability/quality of System", "Improve market share" and "Improve confidence in and perception of system" aim at increasing the revenue for the sales and marketing part of the business structure in the enterprise architecture. "Support improved business processes" enables the software development in the business structures to run smoother.

The system architecture provides a context for the software architecture and includes beside software architecture also hardware and people.

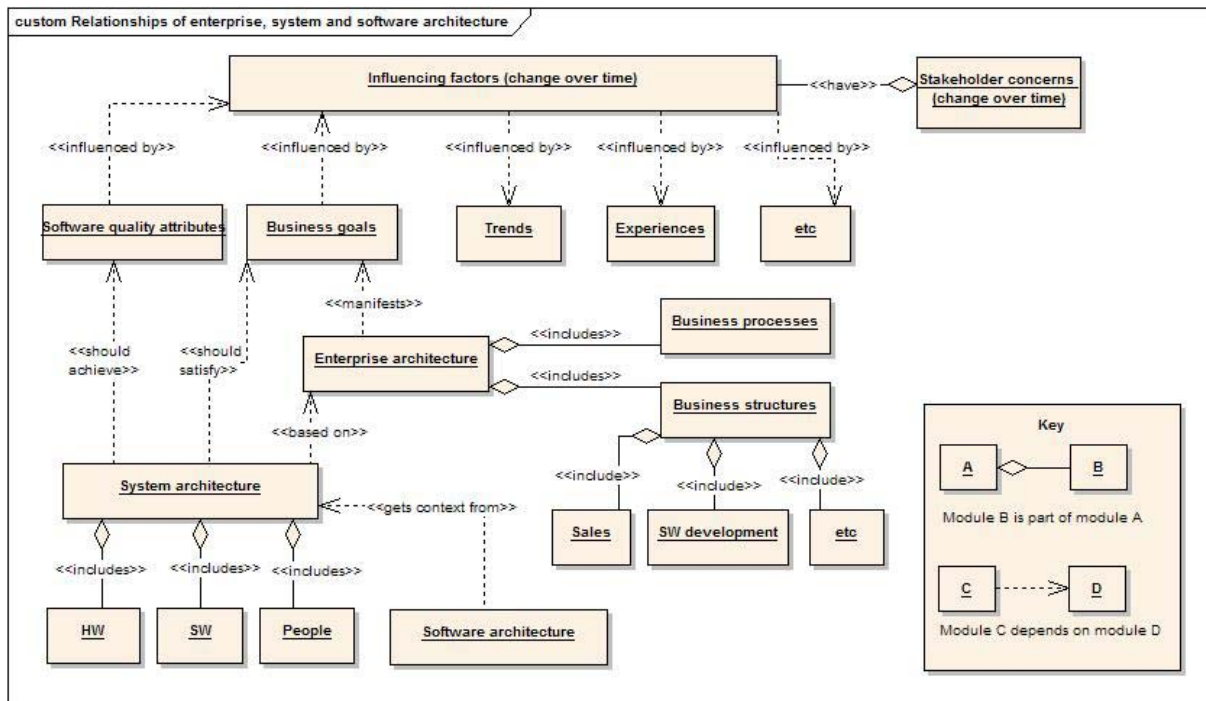


Figure 1. Relationships of enterprise, system and software architecture

4. The IF method

The Influencing Factors method consists of three steps:

- Identify influencing factors,
- Prioritize influencing factors and
- Analyze prioritized influencing factors.

Each step will be described in detail in this section. The steps are best done by the system's software architect and/or software engineer.

4.1 Identify influencing factors

The first step in the IF method collects concerns from different sources like: stakeholder interviews, quality attribute workshop (QAW), discussions with colleagues, search of the business related documents, and personal experiences. From the concerns *influencing factors* are identified. The influencing factor is a factor that affects the architecture design [10].

For identification of an influencing factor the business goal motivations and/or the system quality attribute motivation is noted (if it is stated in the concern). Not all relationships between influencing factors and system quality attributes are possible to extract from its corresponding concern. The software quality attribute influence may be more difficult to trace back to the corresponding concern than the business goal influence. A bottom-up process called affinity diagrams [8] can be used by the project team when classifying the influencing factors' effect on business goals and software attribute qualities. The team members group the influencing factors together by their software quality attribute influence in an affinity sorting process. For each software quality attribute and business goal the influence is divided into:

- Positive impact
- Negative impact
- Requires

A positive impact means an influencing factor which contributes to the achievement of the goal or the implementation of a software quality attribute. An influencing factor having a negative impact means that the influencing factor inhibits the business goal accomplishment or the software quality attribute implementation.

The influencing factor which requires a business goal or software quality attribute requires that specific tactics are used to achieve a certain quality or accomplishment of a business goal. There is a difference between having a positive impact on a system quality attribute and on having a requirement

on it. For example "Support distributed development" requires high degree of "Modifiability" but has no positive impact on the quality. The influencing factor "Implement POSIX compliant software" has a positive impact on modifiability but does not require modifiability, since it is a modifiability tactic itself. The influencing factors are categorized according to their influence on business goals and software quality attributes and can then be entered into the *IF matrix* as shown in Figure 2. The matrix gives a good overview of the influencing factors and especially the trade-offs between them. The influencing factor is entered into the cell which corresponds to its influence on business goals and software quality attribute. In the IF matrix for the two case studies the classified business goals as discussed in [1] and the six quality attributes from [2] are used. But it's possible to use a different set of business goals and software quality attributes that better suits the system being analyzed.

The IF matrix is one way of viewing business goal impact and software quality attribute impact of the influencing factors. If the data was put into a relational database, the user can chose different views of the data than the IF matrix in order to get the best understanding of the impact of and the internal relations between the influencing factors.

4.2 Prioritize influencing factors

In the first step of the IF method the influencing factors were identified and their influences on business goals and software quality attributes were documented in the IF matrix. The second step of the IF method is to identify the prioritized business goals of the system and to extract those influencing factors having a positive impact on the prioritized business goal. It can be easy to identify prioritized business goals, e.g. from interviews with upper management or from the business presentation in a Quality Attribute Workshop. However, sometimes it is more difficult to collect this information, e.g. in a distributed management organization where the system architect has little contact with the business goal responsible management. For this distributed management situation, as is shown in the first case study, it can be that the influencing factors cluster around a positive impact on a specific business goal. The architect can in this case try to verify with the management that this specific business goal is the one prioritized in the organization.

After extracting those influencing factors having a positive impact on the prioritized business goal, step three in the IF method can follow. The results should be verified with the stakeholders so that the

stakeholders having concerns that are not prioritized can get an understanding of the influence of their concerns and why they are not prioritized.

4.3 Analyze prioritized influencing factors

After the influencing factors which have a positive impact on the prioritized business goal(s) are extracted their influence on software quality attributes can be analyzed. The factors are analyzed for their impact on the software quality attributes. For instance, if five influencing factors have a positive impact on the prioritized business goal “Improve market share” those five factors influence on the software quality attributes are analyzed. If all of the five influencing factors require modifiability the architect knows that he/she should implement modifiability tactics and/or patterns in the architecture. This means that the architecture should try to satisfy the concerns related to the influencing factors having a positive impact on the prioritized business goal “Improve market share” and therefore apply modifiability tactics [2] and/or an architectural style [5] and/or pattern [13] with a positive impact on modifiability. If several qualities are required, techniques like the “Cost Benefit Analysis Method” [9] can be used to make cost-oriented decisions on what architectural strategy to apply.

It is also important to analyze the negative impact of the prioritized factors. If the same factors requiring modifiability have a negative impact on performance the architect must take preventive measures not to get a performance drop. The IF matrix shows the internal trade-offs between business goals and internal trade-offs between software quality attributes. It may be that influencing factors having a positive impact on the prioritized business goal “Improve market position” also have a negative impact on the business goal “Reduce total cost of ownership”. In this case the architect can discuss this with the management responsible for the business goal prioritization.

5. Case study 1

The first system on which the architecture team of the software system applied the IF method was a legacy system. The system suffered from an unclear understanding of what business goals the many stakeholders’ concerns were targeting and what software qualities were to be prioritized in the current development of the legacy system. For company confidentiality reasons we cannot publish all descriptions of the influencing factors. But some of the influencing factors are used as examples for clarifying purposes.

5.1 Identify influencing factors

The concerns from stakeholders were collected through interviews, document reading, personal experiences and team discussions. Influencing factors were identified from the concerns and organized according to their influence on business goals and system quality attributes. For some of the factors the influence was not obvious, e.g. there was a factor having both positive and negative impact on the same business goal, the factor IF1.1: *Microsoft Functionality Dependencies*. Using ready produced functionality would make the job easier for the developers, but at the same time introduce costs in terms of licenses and dependencies on Microsoft functionality upgrades. For this factor we have both a positive and a negative impact on the same business goal “Reduce total cost of ownership”. In this particular case we argued that the license cost would be lower than the savings we would get in development cost by using the functionality. Therefore the total impact is a positive impact on the business goal and a negative impact on security due to the introduced external dependency on security patches.

We had several influencing factors having a positive impact on one business goal and a negative impact on another business goal. Especially influencing factors having a positive impact on the business goal “Improve Market Position” tend to have a negative impact on “Reduce Total Cost of Ownership, e.g. IF2.1 “Expand geographical market to China and India”. IF2.1 may involve support for new native languages which results in additional development cost. Figure 2 shows the IF matrix for the first case study.

5.2 Prioritize influencing factors

Since the business goal prioritization was unclear to the project team the business goal focus was clarified by analyzing the influencing factors impact on business goals. The conclusion was that a large majority of the influencing factors was focused on improving market position and confidence in, and perception of the system. We would have expected more focus on “Reduce Total Cost of Development” and “Improve Capability/Quality of System” since the legacy system was ten years old. The focus on “Improving Market Position” and “Improving Confidence in and Perception of the System” was therefore confirmed with the stakeholders. The result was that the following factors were prioritized: IF1.3, IF1.4, IF2.4, IF3.1, IF3.2, IF3.3, IF3.4 and IF5.1. They have a positive

impact on both of the prioritized business goals. These factors will be analyzed in step three of the IF method.

5.3 Analyze prioritized influencing factors

From the matrix in Figure 2 we extracted the prioritized influencing factors and made a list of their impact on software quality attributes, Figure 3. The factors: IF1.3, IF1.4, IF3.2, IF3.3 and IF5.1 required modifiability tactics and/or patterns. The prioritized factors IF2.4, IF3.1 and IF3.3 required usability tactics and/or patterns. The factors: IF3.2 and IF5.1 required testability tactics and/or patterns and the prioritized factor IF3.3 required availability tactics and/or patterns to be implemented in the architecture. Modifiability and usability seemed therefore to be the most important software quality attributes.

Figure 3 also shows that the performance quality was negatively impacted by four of the prioritized influencing factors: IF1.3, IF1.4, IF3.3 and IF3.4. This means we had a trade-off between performance and modifiability/usability/testability.

Figure 4 shows that the business goal “Reduce total cost of development” was negatively impacted by the

prioritized factors: IF1.3, IF1.4, IF3.3, IF3.4 and IF5.1. We therefore had a business goal trade-off between “Reduce Total Cost of Ownership” and “Improving Market Position”/“Improving Confidence in and Perception of the System”.

5.4 Conclusions: Case Study 1

The business goal focus was made visible by the IF matrix. A large majority of the influencing factors had a positive impact on “Improve market position” and “Improve confidence in and perception of the system”. The stakeholders confirmed this business goal focus and we could show that this focus has a strong trade-off with the business goal “Reduce total cost of ownership”.

The IF matrix made internal tradeoffs between business goals visible as well as internal tradeoffs between software quality attributes. A side-effect of the analysis was the discovery that concerns related to the business goal category “Improve confidence in and perception of the system” was strongly correlated to the business goal category “Improve market position”.

Business Goals	Impact	Quality Attributes																	
		Modifiability			Performance			Security			Availability			Testability			Usability		
		Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact
Reduce Total Cost of Ownership	Positive impact	IF1.2, IF3.2																	
	Negative impact	IF1.3, IF1.4, IF2.1, IF2.3, IF3.2, IF3.3, IF5.1	IF3.4, IF4.1, IF4.2	IF3.5	IF2.3, IF3.5		IF1.3, IF1.4, IF2.2, IF3.6, IF3.7, IF4.1, IF4.2	IF2.2, IF3.6		IF2.2, IF3.7				IF1.2, IF3.2				IF3.1, IF3.2	
Improve Capability/Quality of System	Positive impact				IF3.5	IF3.5													
	Negative impact																		
Improve Market Position	Positive impact	IF1.3, IF1.4, IF2.1, IF2.3, IF3.2, IF3.3, IF5.1	IF3.4	IF3.5	IF2.3, IF3.5		IF1.3, IF1.4, IF2.2, IF3.3, IF3.4	IF2.2											
	Negative impact																		
Support Improved Business Processes	Positive impact																		
	Negative impact																		
Improve Confidence in and Perception of the System	Positive impact	IF1.3, IF1.4, IF3.2, IF3.3, IF5.1	IF3.4	IF3.5	IF3.5		IF1.3, IF1.4, IF3.3, IF3.4, IF3.6, IF3.7	IF3.6											
	Negative impact																		

Figure 2. IF Matrix - Case Study 1

Impact on Software Quality Attributes		Modifiability			Performance			Security			Availability			Testability			Usability		
		Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact
Prioritized Influencing Factors	IF1.3	x					x												x
	IF1.4	x					x											x	x
	IF2.4																	x	
	IF3.1																	x	
	IF3.2	x												x				x	
	IF3.3	x					x			x									x
	IF3.4		x				x												x
	IF3.5			x	x														
IF5.1	x												x						x

Figure 3. Quality Attribute Analysis – Case Study 1

Impact on Business Goals		Reduce Total Cost of Ownership			Improve Capability/Quality of System			Improve Market Position			Support Improved Business Processes			Improve Confidence in and Perception of the System		
		Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact
Prioritized Influencing Factors	IF1.3			x				x								x
	IF1.4			x				x								x
	IF2.4							x		x						x
	IF3.1							x								x
	IF3.2	x						x								x
	IF3.3			x				x								x
	IF3.4			x				x								x
	IF3.5			x		x		x								x
IF5.1			x				x								x	

Figure 4. Business Goal Analysis – Case Study 1

6. Case study 2

The system in this case study was a total reconstruction of an old system architecture that was to deliver the same features as the old system, but with additional new functions and qualities. The business goal in this case study was explicitly stated as “Increase sales to year 2009” in contrast to the first case study where the business goal was unclear. For the second case study the IF method was used to prioritize among the multitude of stakeholder concerns and legacy concerns inherited from the old system.

6.1 Identify influencing factors

In order to find the most important stakeholder concerns and their corresponding quality attributes a Quality Attribute Workshop [3], was held. A couple of weeks before the QAW took place we collected requirements in form of use-cases for the new system and legacy requirements from the old system. The business goal requirements were extracted from the business case presentation at the start of the QAW. The voting result was a surprise to the QAW moderators since the top-five scenarios didn’t include the most important legacy quality attribute requirements: performance and availability. The discussion of the result with the participating stakeholders showed that they had voted on the scenarios dealing with new features of the system and ignored the mandatory legacy features. However, by using the IF method we could extract the influencing factors from the concerns related to the legacy requirements as well.

6.2 Prioritize influencing factors

In this case study the prioritization of influencing factors was intensively discussed. Should we only take the ones from the Quality Attribute Workshop prioritization? By following the business goal prioritization from step two in the IF method we included both legacy requirement related influencing factors and top-five QAW scenario related influencing factors. Figure 5 show the overlap of the influencing factors in a Venn diagram. Moreover, Figure 5 also shows that important influencing factors would have been left out if only the influencing factors related to the QAW top-five scenarios would have been analyzed.

The business goal prioritization of influencing factors resulted in the prioritized influencing factors; IF1.1, IF1.2, IF1.3, IF2.1, IF2.2, IF2.3, IF2.4, IF2.5,

IF2.6, IF2.7, IF3.1, IF3.2, IF3.3, IF4.1, IF4.7, IF5.1, IF5.2 and IF5.4.

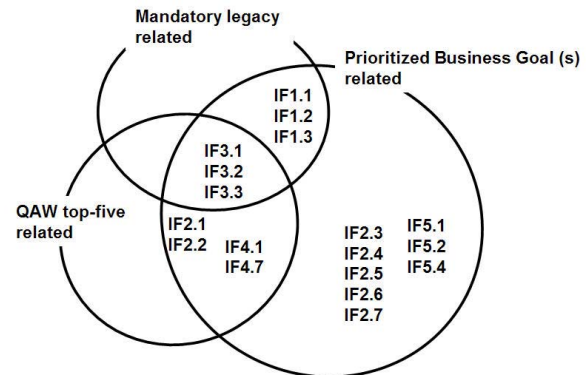


Figure 5. Overlap of influencing factors

6.3 Analyze prioritized influencing factors

In step two we prioritized the influencing factors extracted from concerns related to the prioritized business goal “Improve market position”.

The prioritized influencing factors have requirements on all qualities but the majority of the influencing factors require modifiability and usability, Figure 6. Most of them have a tradeoff with the performance quality. The architecture can not be constructed only to satisfy the requirements on modifiability and usability without regarding the performance requirement of the IF1.2 “Implement same performance as today”. The IF 1.2 has a strong legacy requirement on performance and actually drives the architecture.

Figure 7 shows that nine of the eighteen prioritized influencing factors have a negative impact on the business goal “Reduce total cost of ownership”. That is, the business goal “Improve Market Position” has a trade-off with the business goal “Reduce total cost of ownership”.

6.4 Conclusions: Case Study 2

In this second case study the IF method was a necessary complement to the QAW. The stakeholders who participated in the QAW put their votes on scenarios describing new product functionality and new product qualities. Therefore all legacy requirements and prioritized business goals didn’t get into the top-five scenario ranking.

One question we had regarding step two in the IF method was if it was sufficient to use prioritized business goal(s) as selection criteria for the prioritization of IFs. Case study two showed us that by

using the prioritized business goal(s) as criteria we covered IFs extracted from mandatory legacy requirements concerns and from the QAW top-five scenario concerns. This could have been the case since mandatory legacy requirements are a part of the business goal “Improve market position” which has the

subcategory “Expand or Retain market share”. The subcategory implies that the functionality and qualities of the system must be retained or improved.

The result from the QAW was used as input to the IF method and the output from the IF method as input to the ADD method [6] and the USAP method [7].

Impact on Software Quality Attributes			Modifiability			Performance			Security			Availability			Testability			Usability			
			Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	
Prioritized Influencing Factors	Mandatory Legacy Requirements related	IF1.1																	x		
		IF1.2			x	x															
		IF1.3					x														
		IF3.1	x																		
		IF3.2																			
		IF3.3																			
	QAW top-five scenarios related	IF2.1																			x
		IF2.2																			x
		IF4.1																			
		IF4.7	x																		
	Positive impact on prioritized business goal	IF2.3	x																		x
		IF2.4	x																		x
		IF2.5																			x
		IF2.6																			x
		IF2.7																			x
IF5.1		x																			
IF5.2		x																			
IF5.4																					

Figure 6. Software Quality Attribute Analysis – Case Study 2

Impact on Software Quality Attributes			Reduce Total Cost of Ownership			Improve Capability/Quality of System			Improve Market Position			Support Improved Business Processes			Improve Confidence in and Perception of the System						
			Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact	Requires	Positive impact	Negative impact				
Prioritized Influencing Factors	Mandatory Legacy Requirements related	IF1.1																			
		IF1.2																			x
		IF1.3																			
		IF3.1																			x
		IF3.2																			x
		IF3.3																			x
	QAW top-five scenarios related	IF2.1																			x
		IF2.2																			x
		IF4.1																			
		IF4.7																			x
	Positive impact on prioritized business goal	IF2.3																			
		IF2.4																			
		IF2.5																			x
		IF2.6																			x
		IF2.7																			
IF5.1																					
IF5.2																					
IF5.4																					

Figure 7. Business Goal Analysis – Case Study 2

7. Conclusions

The IF method extracts influencing factors from stakeholders' concerns. The influencing factor is a factor that affects the architecture design. The influencing factors' impacts on software quality attributes and business goals are analyzed. In the two case studies the gathering of concerns from stakeholders took about a person week and the contribution of each stakeholder was approximately two hours of interviews for those that couldn't participate in the one day quality attribute workshop.

From the two case studies we concluded that for a skilled architect with business goals understanding and software quality attributes skills, the IF analysis of the concerns should take no longer than a day or two.

Changes in business goal focus during the life-time of the software system means that new influencing factors are added. The added influencing factors and their impacts may be added to the existing ones in a relational database and the view of the impact can be presented in many ways, e.g. in the IF matrix format.

In the IF method the business goal prioritization is central. It is the prioritized business goals that controls what concerns will be prioritized.

One difficulty in the IF method is the categorizing of impact on business goals and software quality attributes in step two. This is one difficulty the IF method shares with methods described in [3], [4] and [6].

8. Future work

We will investigate the possibility to apply the IF method before a QAW and use the result to understand the effect of the stakeholders' concerns and present this to the stakeholders before starting the QAW. This might help the stakeholders to make more focused voting decisions.

Moreover, it would be interesting to look deeper into the correlation between business goals and software quality attributes. In our two case studies we have seen a high correlance between "Usability" and "Improve Confidence in and Perception of the System" and between "Usability" and "Improve Market Position".

Finally more research in the field of influence of concerns on business goals and software quality attributes will make step two in the IF method more precise.

9. References

- [1] L. Bass, and R. Kazman, "Categorizing Business Goals for Software Architectures", *TECHNICAL REPORT CMU/SEI-2005-TR-021 ESC-TR-2005-021*, 2005.
- [2] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice, Second Edition*, Addison-Wesley, Boston, 2003.
- [3] M. Barbacci, R. Ellison, A. Lattance, J. Stafford, C. WeinStock, C, and W. Wood, "Quality Attribute Workshops, 3rd Edition", *TECHNICAL REPORT CMU/SEI-2003-TR.016*, Pittsburgh, PA, 2003.
- [4] P. Clements, R. Kazman, and M. Klein, *Evaluating Software Architectures, Methods and Case Studies*, Addison-Wesley, Boston, 2002.
- [5] D. Garlan, and M. Shaw, *Software Architecture: Perspectives on an emerging discipline*, Prentice-Hall, Inc., 1996.
- [6] R. Wojcik, F. Bachmann, L. Bass, P. Clements, P. Merson, R. Nord, and B. Wood, "Attribute-Driven Design (ADD), Version 2.0", *TECHNICAL REPORT CMU/SEI-2006-TR-023 ESC-TR-2006-023*, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006
- [7] L. Bass, and B. E. John, "Linking usability to software architecture patterns through general scenarios", *The Journal of Systems and Software* 66, 2003, pp. 187-197.
- [8] H. Beyer, and K. Holtzblatt, *Contextual Design*, San Francisco, CA, Morgan Kaufmann Publishers, Inc., 1998.
- [9] R. Kazman, J. Asundi, M. Klein, "Making Architecture Design Decisions: An Economic Approach", *TECHNICAL REPORT CMU/SEI-2002-TR-035 ESC-TR-2002-035*, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002.
- [10] C. Hofmeister, R. Nord, and D. Soni, *Applied Software Architecture*, Boston, Addison-Wesley, 1999.
- [11] "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems", *IEEE Computer Society*, 2000, pp. i-23.
- [12] N. G. Leveson, "Intent Specifications: An approach to Building Human-Centered Specifications" *IEEE Transactions on Software Engineering, Volume 26, No 1*, 2000.
- [13] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, *Pattern-oriented Software Architecture A System of Patterns, Volume 1, First Edition*, Wiley, 1996.