MIND THE GAPS in the requirements flow!


by Elizabeth Bjarnason
Department of Computer Science, Lund University, Sweden

We use software all the time and expect it to do what we want. That the auto-corrector of our mobile phone chooses the word we mean, that the Satnav avoids blocked roads, and that the internet bank helps us pay our bills to the intended recipient.

Software products become smarter and more competent. A smartphone today has as much computing power and memory as a personal computer of 10 years ago and contains almost as much functionality. At the same time we expect it to work without hassle and help us in our everyday lives.

But, which of the users many expectations and requirements should the product support? And, how should they be conveyed to the hundreds of engineers involved in producing the software? The different parts of an organization that develop software need to be coordinated and have a common
view of what the product should be in order to work towards a joint goal.

**Misunderstandings are expensive**

Misunderstanding of the meaning of the product requirements causes problems and can result in a product with a different behaviour than what the customer had expected. At best these problems are caught before the product reaches the customer but additional work is still required to correct the misunderstandings.

The aim of our research is to improve the precision and efficiency of software development by locating and avoiding communication gaps and flaws in the coordination between engineers. Specifically, my thesis focuses on methods that can identify potentially troublesome gaps and thereby support organisations that develop software in strengthening the coordination and alignment between requirements engineering and testing activities.

**Complex Alignment**

The requirements for a new software product gradually evolve by balancing the expectations of the customer, the company’s business strategy and plans with the technical possibilities and development costs. This is a complex process that persists throughout the development process. In this process, many of the details are designed and decided by the engineers themselves. To coordinate and align such continuously evolving requirements requires good collaboration and communication as well as competence, routines and tools to manage and document this information.

A weak alignment between the product requirements and the engineers can lead to producing software that deviates from the agreed requirements and that does not meet the customers’ expectations. Correcting this then requires changes to the software, which can lead to both delays and increased development costs.

If the test engineers also have misinterpreted the requirements there is a risk that the product contains such deviations when it reaches the customer. The product may then need to be recalled from the market that incurs increased costs and can also affect customer confidence in the manufacturer and reduced brand value.

**Visualizing Gaps**

These communication gaps in the requirements flow are the focus of my research. The aim is to provide methods that visualize the gaps and thereby support development projects in realizing where there are flaws in the

![A timeline used in the EBTR method.](image-url)
communication flow and how these gaps can be bridged.

The thesis describes a new method for supporting group reflections (the EBTR method) of a completed project. Through such reflections the engineers can gain new insights into strengths and weaknesses of the interactions between project members. The method provides the project team with a common view of the project through visualizing passed project events along a timeline. This timeline supports an objective discussion within the group based on facts.

In addition, we have developed the Gap Finder, a method for measuring various distances between requirements and testing activities. The gaps between engineers and between documents indicate potential misunderstandings in the requirements communication. For example, large differences (or gaps) between the requirements engineer and the test engineers knowledge of the product and its customers incurs a risk for misunderstandings of the requirements. This risk can be mitigated by introducing new practices such as the requirements engineer testing parts of the software functionality.

In conclusion, time and money can be saved by developing software according to a common understanding of the requirements. The amount of software will continue to increase and as consumers we require more functionality while our tolerance for badly functioning software decreases. The visualization of distances between customer requirements and the understanding of these within a development project can help engineers to identify and bridge gaps in the communication of these requirements. The alignment within a development organization can thereby be improved. Late changes and rework of the software can be avoided by implementing the ‘right’ software, i.e. according to a correct understanding of the requirements. The productivity of the software development organization is thus be improved.

**Insight Strengthens Alignment**

The EBTR method for group reflections and the Gap Finder have both been applied to several software development projects and evaluated by a combination of questionnaires and focus groups. The visualization of project events with EBTR initiated relevant and valuable group discussions. In particular, the test engineers gained new insights concerning the requirements flow within the project. The Gap Finder identified a number of gaps in the requirements flow of a development team and suggested a number of improvement practices which lead to strengthened alignment within the project team.

In conclusion, time and money can be saved by developing software according to a common understanding of the requirements. The amount of software will continue to increase and as consumers we require more functionality while our tolerance for badly functioning software decreases. The visualization of distances between customer requirements and the understanding of these within a development project can help engineers to identify and bridge gaps in the communication of these requirements. The alignment within a development organization can thereby be improved. Late changes and rework of the software can be avoided by implementing the ‘right’ software, i.e. according to a correct understanding of the requirements. The productivity of the software development organization is thus be improved.

*Insight Strengthens Alignment*

The EBTR method for group reflections and the Gap Finder have both been applied to several software development projects and evaluated by a combination of questionnaires and focus groups. The visualization of project events with EBTR initiated relevant and valuable group discussions. In particular, the test engineers gained new insights concerning the requirements flow within the project. The Gap Finder identified a number of gaps in the requirements flow of a development team and suggested a number of improvement practices which lead to strengthened alignment within the project team.

In conclusion, time and money can be saved by developing software according to a common understanding of the requirements. The amount of software will continue to increase and as consumers we require more functionality while our tolerance for badly functioning software decreases. The visualization of distances between customer requirements and the understanding of these within a development project can help engineers to identify and bridge gaps in the communication of these requirements. The alignment within a development organization can thereby be improved. Late changes and rework of the software can be avoided by implementing the ‘right’ software, i.e. according to a correct understanding of the requirements. The productivity of the software development organization is thus be improved.

*Insight Strengthens Alignment*

The EBTR method for group reflections and the Gap Finder have both been applied to several software development projects and evaluated by a combination of questionnaires and focus groups. The visualization of project events with EBTR initiated relevant and valuable group discussions. In particular, the test engineers gained new insights concerning the requirements flow within the project. The Gap Finder identified a number of gaps in the requirements flow of a development team and suggested a number of improvement practices which lead to strengthened alignment within the project team.

In conclusion, time and money can be saved by developing software according to a common understanding of the requirements. The amount of software will continue to increase and as consumers we require more functionality while our tolerance for badly functioning software decreases. The visualization of distances between customer requirements and the understanding of these within a development project can help engineers to identify and bridge gaps in the communication of these requirements. The alignment within a development organization can thereby be improved. Late changes and rework of the software can be avoided by implementing the ‘right’ software, i.e. according to a correct understanding of the requirements. The productivity of the software development organization is thus be improved.

*Insight Strengthens Alignment*

The EBTR method for group reflections and the Gap Finder have both been applied to several software development projects and evaluated by a combination of questionnaires and focus groups. The visualization of project events with EBTR initiated relevant and valuable group discussions. In particular, the test engineers gained new insights concerning the requirements flow within the project. The Gap Finder identified a number of gaps in the requirements flow of a development team and suggested a number of improvement practices which lead to strengthened alignment within the project team.

In conclusion, time and money can be saved by developing software according to a common understanding of the requirements. The amount of software will continue to increase and as consumers we require more functionality while our tolerance for badly functioning software decreases. The visualization of distances between customer requirements and the understanding of these within a development project can help engineers to identify and bridge gaps in the communication of these requirements. The alignment within a development organization can thereby be improved. Late changes and rework of the software can be avoided by implementing the ‘right’ software, i.e. according to a correct understanding of the requirements. The productivity of the software development organization is thus be improved.

*Insight Strengthens Alignment*

The EBTR method for group reflections and the Gap Finder have both been applied to several software development projects and evaluated by a combination of questionnaires and focus groups. The visualization of project events with EBTR initiated relevant and valuable group discussions. In particular, the test engineers gained new insights concerning the requirements flow within the project. The Gap Finder identified a number of gaps in the requirements flow of a development team and suggested a number of improvement practices which lead to strengthened alignment within the project team.

In conclusion, time and money can be saved by developing software according to a common understanding of the requirements. The amount of software will continue to increase and as consumers we require more functionality while our tolerance for badly functioning software decreases. The visualization of distances between customer requirements and the understanding of these within a development project can help engineers to identify and bridge gaps in the communication of these requirements. The alignment within a development organization can thereby be improved. Late changes and rework of the software can be avoided by implementing the ‘right’ software, i.e. according to a correct understanding of the requirements. The productivity of the software development organization is thus be improved.

*Insight Strengthens Alignment*

The EBTR method for group reflections and the Gap Finder have both been applied to several software development projects and evaluated by a combination of questionnaires and focus groups. The visualization of project events with EBTR initiated relevant and valuable group discussions. In particular, the test engineers gained new insights concerning the requirements flow within the project. The Gap Finder identified a number of gaps in the requirements flow of a development team and suggested a number of improvement practices which lead to strengthened alignment within the project team.

In conclusion, time and money can be saved by developing software according to a common understanding of the requirements. The amount of software will continue to increase and as consumers we require more functionality while our tolerance for badly functioning software decreases. The visualization of distances between customer requirements and the understanding of these within a development project can help engineers to identify and bridge gaps in the communication of these requirements. The alignment within a development organization can thereby be improved. Late changes and rework of the software can be avoided by implementing the ‘right’ software, i.e. according to a correct understanding of the requirements. The productivity of the software development organization is thus be improved.