

Table of Contents

<i>What is problem analysis?</i>	1
<i>Why do we analyze design problems?</i>	1
<i>How do you analyze design problems?</i>	2
<i>Deliverables</i>	2
<i>However...</i>	3
<i>References</i>	3

Problem Analysis

Understanding - *really* understanding - a design problem is [half the battle](#) in finding a good design solution.

What is problem analysis?

Problem analysis is a set of analytic tasks meant to increase the designers' understanding of an unbalanced situation, for the sake of designing a change to the situation that will have better [balance](#).

In other words, problem analysis involves developing a set of [requirements](#) that will be satisfied by any suitable design intervention, and *only* by suitable design interventions.

With reference to Simon's definition of designing, the requirements will quantify what we mean by a "preferred situation" that must be achieved.

Why do we analyze design problems?

Situations are problematic only for people - nature doesn't have any "problems." To understand a situation, you have to understand the *people* and other [agents](#) who find the situation dispreferred.

It is vitally important that your understanding of the problem be properly documented, because:

1. you don't want the problem to "drift" over time and become something different because of shortcomings in your memory or thinking;
2. you can think better about problems by "externalizing" them;
3. you need a reference document describing the problem for the entire team; and
4. in "real life," formal problem specifications often become part of legal contracts between companies.

If you cannot properly analyze a design problem, then you are quite likely to come up with a "bad" design - where a "bad" design is one that doesn't address the [imbalances](#) in a [situation](#). Since we learn best from our mistakes, it is quite useful to look at bad designs and think about what's wrong with them and how they might be fixed.

Some examples of bad designs (or badly analyzed design problems) include:

- The [staff washroom in EPH-306D](#).
- A [security system developed for the US Department of Homeland Security](#).
- The [SSRMS](#).
- A [coffeemaker case study](#).

Finally, there is one more reason for performing a detailed and well-documented problem analysis: it **helps "synchronize" the thinking of all the team members**. At the very outset of a design project,

the problem is almost invariably defined in quite vague terms. Everyone on the team has a different experience base, different attitudes and perspectives, different amounts of knowledge, etc. In the situation of a vague problem, this means, different team members will very likely end up with different *mental models* of the problem. This is a problem, because it means each team member - if left to their own devices - will address a slightly different problem than every other team member. This will end badly for everyone.

By conducting problem analysis in a team setting, all team members will (if it's done right) end up agreeing quite precisely on the actual imbalance(s) that have to be addressed. That is, all team members will have the same mental model of the problem - and so the resulting design intervention is that much more likely to address the imbalances well.

How do you analyze design problems?

Assuming you have completed the [project initialization](#) phase, there are two main steps to problem analysis:

1. Quantify what specific improvements will constitute a “preferred situation”, and
2. Capture and structure all the problem specification information in the form of [requirements](#).

Developing a set of requirements will probably induce questions (a) that you had not yet thought of, yet (b) the answers to which will improve the requirements. This results in an [iterative](#) process, composed of alternating phases of exploration, during which more information is gathered or identified, and analysis, during which information is pared down, made more concise and precise, and formalized.

The basic steps of the process (per the [Design Roadmap](#)) are as follows.

See [scoping](#) for a general introduction.

1. form [team](#)
2. hold [kickoff meeting](#)
3. conduct a [situation scan](#)
4. establish [situated use cases](#)
5. establish [human factor capabilities](#)
6. define [personas](#)
7. develop [requirements](#)
8. perform [requirements validation](#)
9. prepare a [PRS](#)

2016.08.01 19:40 · Fil Salustri

Deliverables

The formal deliverable of the problem analysis stage is a [product requirements specification](#).

However...

Many [imbalances](#) do not relate directly to engineering concerns. Since our responsibilities and obligations are only to the engineering aspects of a design problems, we need to be able to pick out those technical aspects and focus on them. However, the technical aspects of a specific situation are coupled to other non-technical aspects in the broader societal situation. *This is why we cannot ignore completely the non-technical aspects of situations.*

Indeed, our proposed resolutions for the technical aspects of unbalanced situations are *constrained* by the non-technical issues, because we, as engineers, have no control over the non-technical issues - that is, in the same way that we as engineers are supposed to have control over the technical issues, we must accept that there are other kinds of experts who have control over the non-technical aspects.

Knowing where to draw the boundaries between the aspects on which you need to focus and all the other aspects can be quite vague. There may well be some aspects about which your team cannot decide. In such cases, do not hesitate to seek guidance from your instructors.

Also, even with all the best intentions and meticulous execution of this stage, it is *not* guaranteed that everyone will agree. The best one can expect, usually, is “informed compromise” [\[AEF00\]](#) about the problem that has to be solved. Managing those compromises will become a key element of the process as the design proceeds.

Remember, it is neither necessary nor absolutely beneficial for every team member to understand every aspect of the design problem [\[AEF00\]](#). In the general case, there is simply too much to know¹⁾.

References

AEF00. a, b E. Arias, H. Eden, G. Fischer, A. Gorman and E. Scharff 2000. **Transcending the individual human mind—creating shared understanding through collaborative design.** *ACM Transactions on Computer-Human Interaction*, 7(1):84–113. ([link](#))

[process](#), [analysis](#), [balance](#)

¹⁾

Example: the Boeing 777 is such a complex aircraft that it is often described as *four million parts flying in close formation* because no single person - of the over 5,000 engineers who worked on it - have complete knowledge about it.

From:

<https://deseng.ryerson.ca/dokuwiki/> - **DesignWIKI**

Permanent link:

https://deseng.ryerson.ca/dokuwiki/design:problem_analysis



Last update: **2020.03.12 13:30**