# Clinical software product improvement based on incident reports

Tesfay Teame, Regionalt Senter for Klinisk IKT

Tor Stålhane, NTNU / IDI

Øystein Nytrø, NTNU / IDI

# Contents

- On reactive improvement
- Selection of data analysis approach
  - Five Whys
  - Why - Because
  - What we selected – and why
- Data analysed
  - Error report example
  - Five Whys example
- Problem analysis
  - First level problems
  - Detailed problem categories
- Conclusions
- Where do we go from here

# Why reactive improvement

Our presentation focuses on reactive improvement.

Reactive improvement will contribute to better and quicker adaptation of  a health service that is in continuous change, e.g. EPJ.

Precise healthcare requires that

- we consider all relevant circumstances when making decisions
- adapt the IT-systems to the users' needs – all the time

# Reactive improvement – 1

Reactive improvement

- is different from incremental, iterative or agile development.
- implies that the information system is not in development or in design but is considered to be fully deployed, operational and in use by the customer organization.
- is *in situ*, *managed by the owner/operator*, *as part of a health organization*.

# Reactive improvement – 2

Reactive improvement, based on helpdesk reports, relies on deep organizational and system insight to be able to

- isolate root causes
- propose possible remedies.

The health record ecosystem has hundreds of modules, user interface components and devices intertwined with organizational processes and role responsibilities.

It is natural to categorize errors according to four main dimensions:

- Apparent situation of discovery/organizational context
- Type of system malfunction or error
- Apparent system function or module
- Seriousness or risk of patient harm.

# Reactive improvement – 3

- Each reported incident is an opportunity to learn how to avoid repeating it.
- This way of improving a complex system has been used with success in several branches of industry – e.g., the aviation industry
- There exist a lot of methods but we will just look at two of them – “Five Whys” and “Why Because Analysis”. We will use a set of errors reported to a help-desk as input data.

# Selection of approach

How to do a root cause analysis will depend on the following factors:

- What kind of data do we have
- Who shall participate in the analysis
- How was the data collected – or more precisely – how much control did we have

The first two points determine the method used while the last point will decide threats to validity.

In our case

- we have only textual data
- the purpose is to extract knowledge.

We have chosen to discuss two methods in more details

- “Five Whys”
- “Why – Because”.

# Five Whys – 1

The idea of “Five Whys” is to identify the problem, ask why it did occur and keep on asking “why” five times:

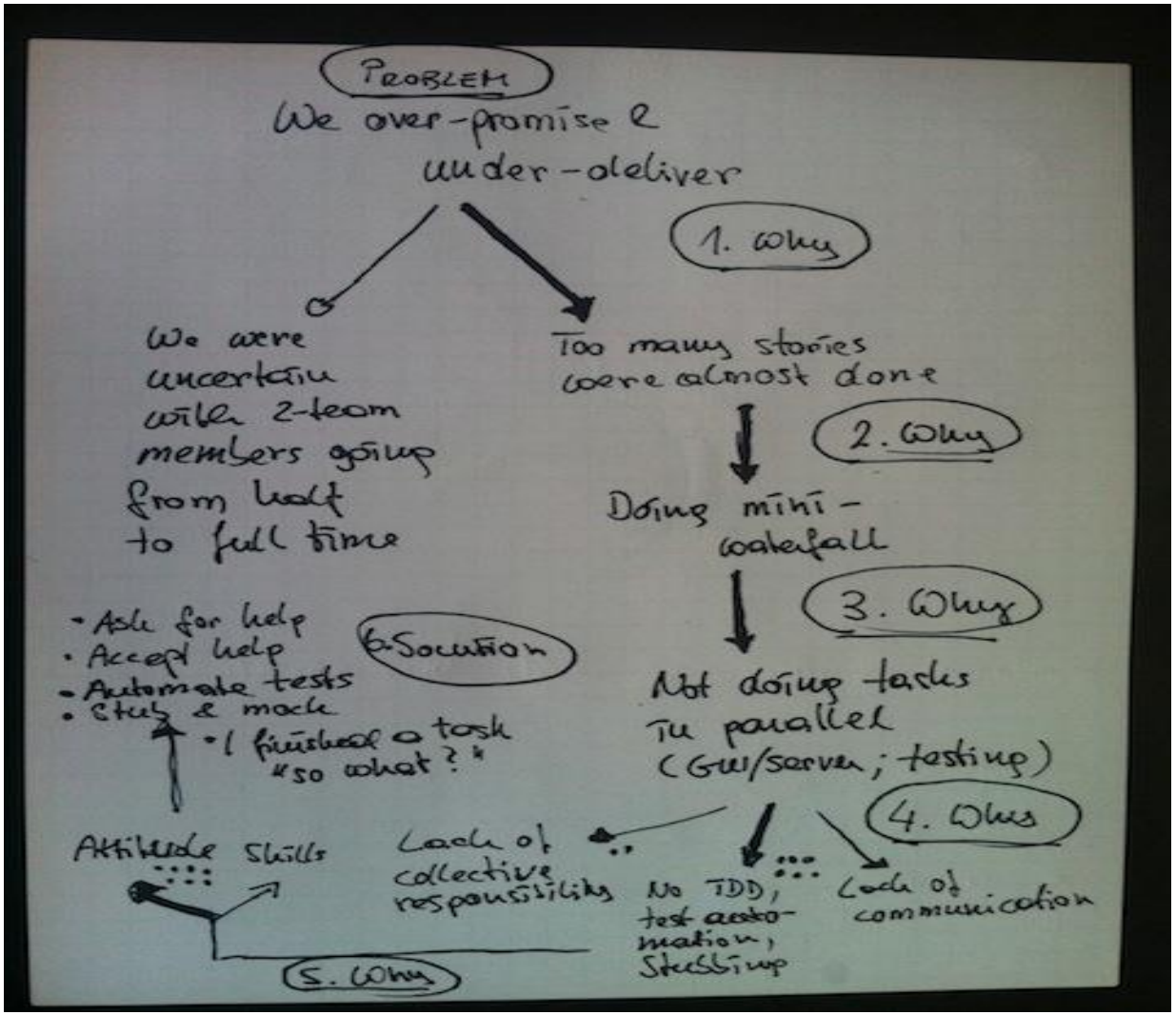
1. Identify the problem – what are we trying to achieve. Spend some time here. Focus on the root cause.
2. “Why did this happen?” Identify all the causes you can think of.
3. For each of the causes identified in step 2 ask “Why did this happen?”
4. Repeat steps 2 and 3 five times. By this stage, we should have identified all relevant root causes.
5. Identify solutions and countermeasures to the causes identified in steps 2 and 3.

# Five Whys – 2

Following the five steps identified earlier, the whole process can be documented using the table below.
When we have reached the last step in the table, we need to suggest actions that will remove the problem

| Step | Reason | Why? |
|---|---|---|
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| Root cause | | |

# Five Whys – 3




A Five Whys tree. We get a tree when one or more Whys has several answers.

# Why - Because

1. Collect information
2. Determine facts.
   a) Go through all collected information
   b) Split it up into single events.
   c) Document all assumptions
3. Make a list of all relevant facts
4. Create a WBA list, based on facts from step 3. This step will generate a set of pairs of facts - A causes B
5. Build a time line by assigning an actor and a time to each event.
6. Identify the mishap – the top node in the WBA graph. This is the event that causes the accident
7. Generate the WBA tree using the WBA list and the time line. Start with the mishap.
   a) For all nodes in the graph, add child nodes based on the WBA list
   b) Repeat a) until all entries in the Why-Because list are used
8. Go through the WBA tree to check for errors, misunderstandings and incompleteness.

# Why “Five Whys”

Both methods end up with a cause-consequence tree.

WBA has a high degree of formality which will help to reduce the degree of subjectivity of the results.

There are, however, important reasons to not use this method

- The amount of work needed makes WB impractical when analysing a large set of events – e.g. a set of error reports.
- Experience from other areas – e.g. risk analysis – indicates that
  - We in most cases should base our decisions on the two or three most important factors
  - Adding more factors only water down the important ones.

We settle for the Five Whys analysis method.

# Data analysed – why these data

The data used are collected from a helpdesk at Sykehuspartner.

Government owned hospitals in the South-East health region report errors related to Clinical ICT to Sykehuspartner which must find the cause and solve the problem within an agreed period of time.

More than 13000 error reports with criticality 1B – errors related to Clinical applications – are reported from January 2013 to July 2014.

We have chosen to examine error reports that are related to DIPS EHR since this application is widely used.

We analysed 1618 error reports. The errors are reported from nine hospitals in the south-east health region

# Report example

**Hospital**: XXHF

**Title**: The application (DIPS) is running too slow

**Description**: Users can sign in. But the application freezes right after it started running. The problem affects multiple (all?) users. The problem is probably related to Secured Zone.

**Solution:** 09:55 hours. The Operation Centre discovers the problem.

- No one notified the Centre about the problem. Tried to figure out the scope.
- Contacted Operation coordinator at the helpdesk and queue coordinator at clinical application unit. They didn't realize how big the problem is.
- Queue coordinator from Technical System operation contacted technicians from applications servers and asked if any server is down.
- Y from helpdesk mentioned that a print server is down and it why DIPS freezes.
- We degraded the criticality from 1A to 1B, since the problem is related to a print server and not in DIPS itself.

# “Five Whys” example

| Step | Reason | Why? |
|---|---|---|
| **1** | DIPS freezes right after it starts running | Multiple users didn’t access DIPS |
| **2** | The application didn’t access a default printer | Why did DIPS freezes right after it starts running |
| **3** | The print server was down | Why the application didn’t access a default printer |
| **4** | Because DIPS checks a default printer for the user when the user sign in | Why a down print server make DIPS freeze/slow |
| **5** | DIPS is programmed to use 30sec to check one default printer, before it checks the next one | Why this checking makes DIPS freeze/slow |
| **Root cause** | Dependency between DIPS and a default printer to start up the application | |

# First-level problems discovered

Two general findings

- Poor data quality: 33% of the reported errors did not have detailed information about what happened and done to solve the problem. Example:
  - 207 reports refer to a telephone call. The case is closed after a telephone call with the user
  - 304 reports have only “the problem is solved” in their solution parameter without mentioning what have been done to solve the problem.
- Servers that were down and not discovered on time - in most cases the technicians discovered that the server is down after one or several users have reported an error.

# Problem categories

| Problem category | Number of failures | Percentages |
|---|---|---|
| **Bad manual procedures** | 9 | 35 |
| **Configuration problems** | 6 | 23 |
| **Lack of resources** | 5 | 19 |
| **Deficiencies in the control system** | 3 | 11 |
| **Unknown cause** | 2 | 7 |
| **Long delays in National Population Register** | 1 | 5 |
| **Sum** | 26 | 100 |

The main problem categories are related to the users' organization

- Bad manual procedures
- Configuration problems
- Lack of resources, e.g. for training.

# Conclusion

- Norwegian hospitals are often unprepared for the challenges related to introducing information technology.
  - Too few resources are allocated.
  - The necessary manual procedures are not good enough and are not tested thoroughly.
- The problems are solvable but require that the hospital administration understand them and want to do something about them.
- Bad data quality makes a large part of the reported incident impossible to analysis => the hospitals throw away improvement opportunities.
- The incident data quality will improve when the personnel see that it is used to improve the systems and the service they provide.

# Where do we go from here

Both the health sector and software industry should regard detected failures as an opportunity for continued improvement.

Improvement requires detailed reporting, open communication and analytical resources.

All problems should be analysed using e.g. “Five Whys” and the identified root causes should be removed or at least reduced.