

# Optimized Fault Handling in Global Software Development

A Case Study of Six Sigma applied in Software Development

ICSP 2008, Leipzig, 2008-05-11

Dirk Macke, Ericsson Deutschland GmbH

Tihana Galinac, University of Rijeka

# Six Sigma – The DMAIC Process



Understanding the problem or opportunity and its financial impact

Develop and execute appropriate data collection method

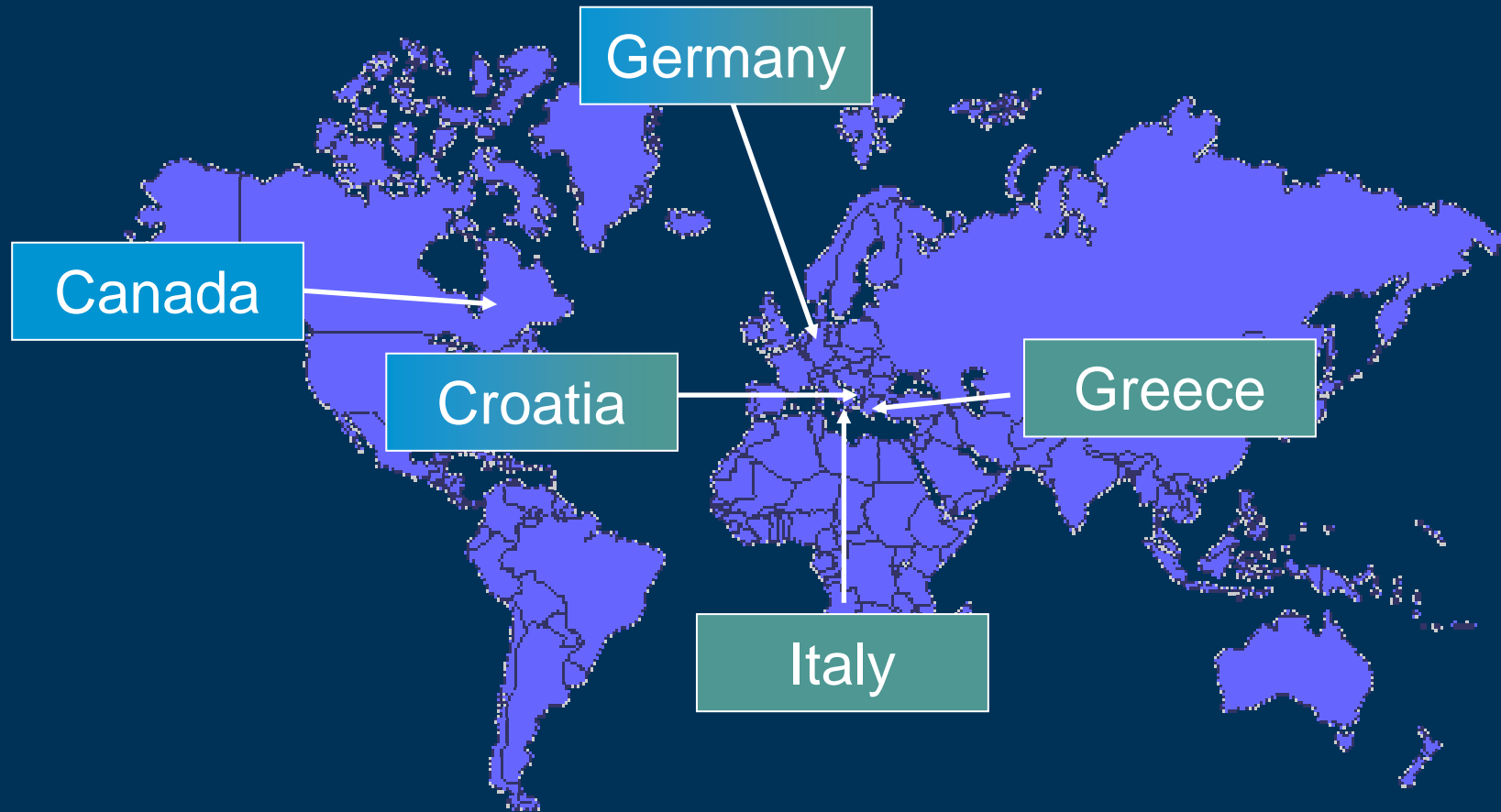
Find the root causes

Generate and implement solutions

Ensuring the results



# Ericsson Mobile Switching Center Global Software Development Organization



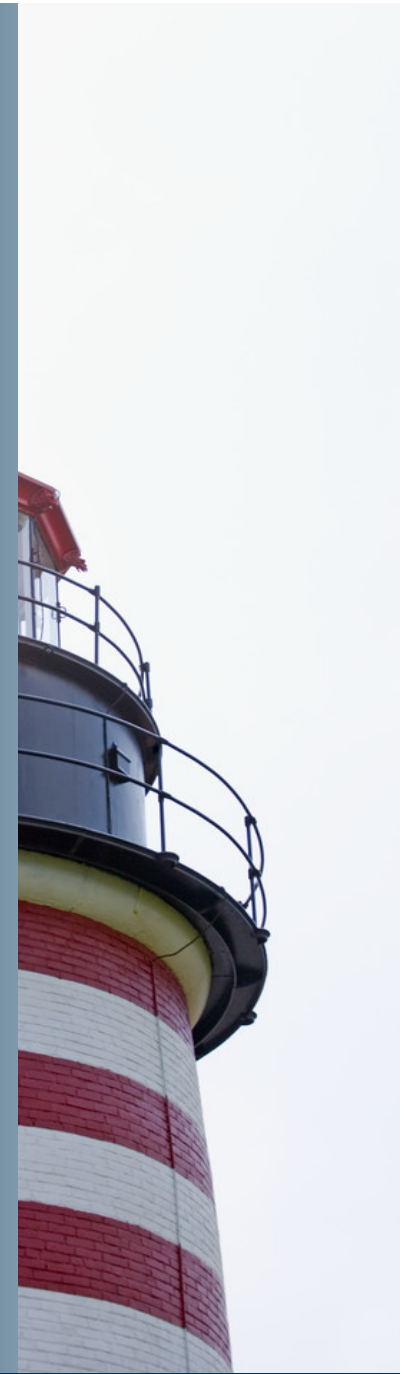
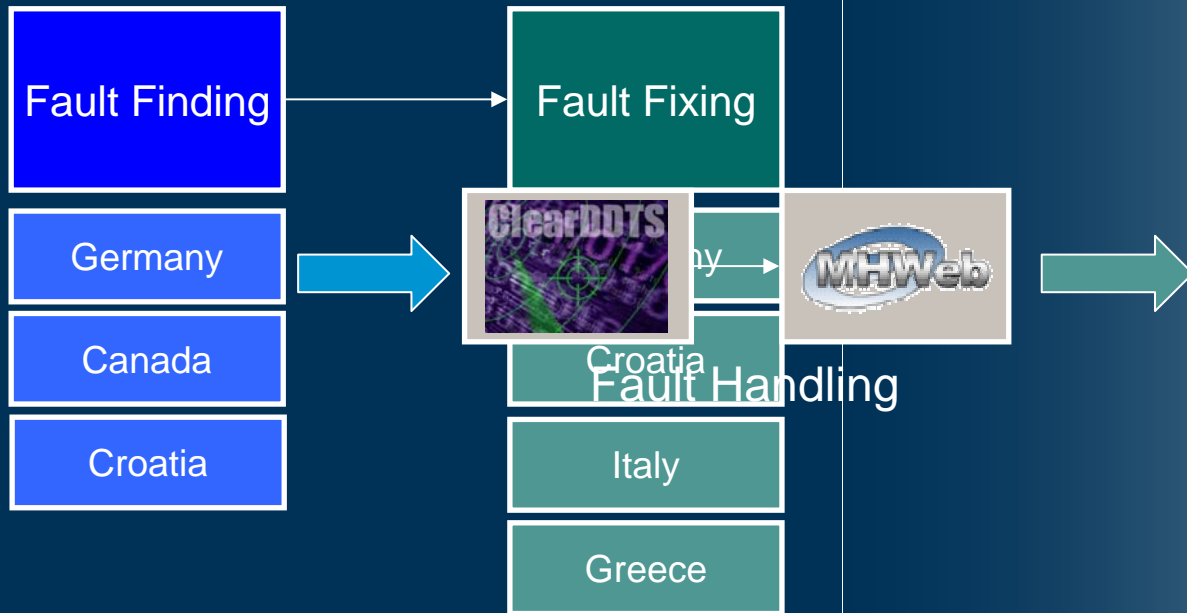
- Global Design Organizations
- Global Network Integration & Verification Organizations

Only main sites subject to this case study shown.

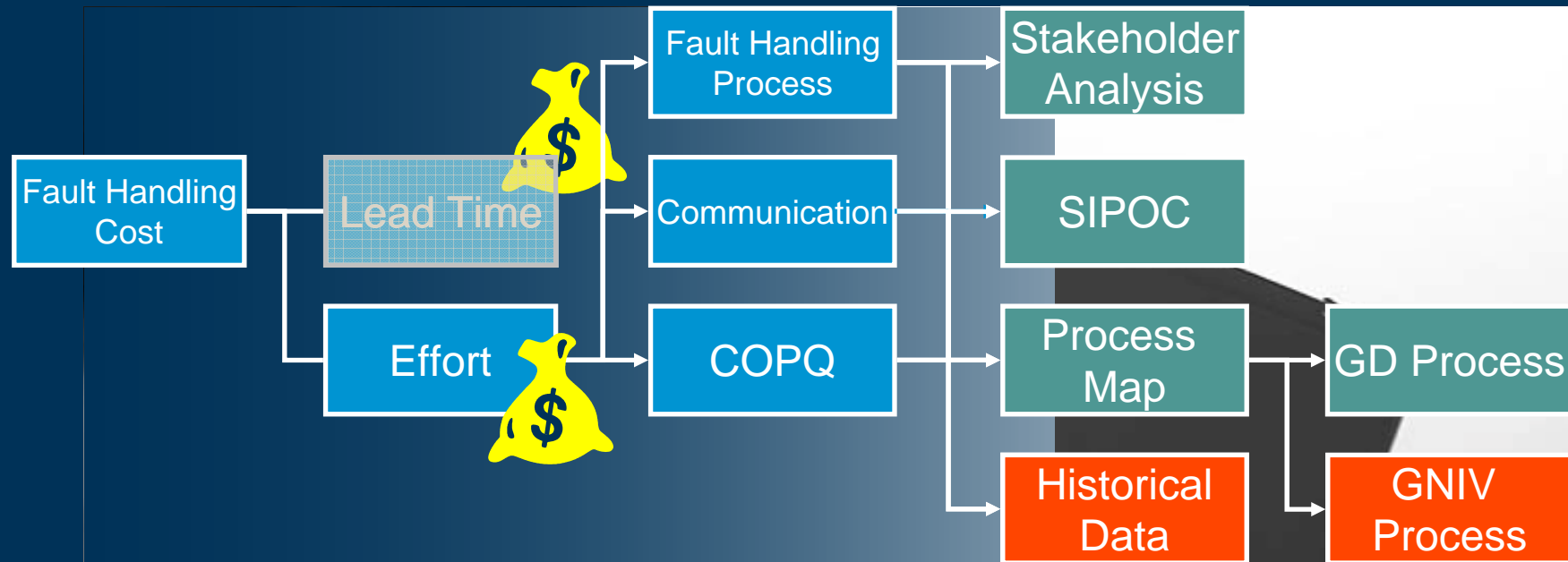
# Background

Global Network  
Integration & Verification  
(GNIV)

Global Design  
Maintenance (GD)

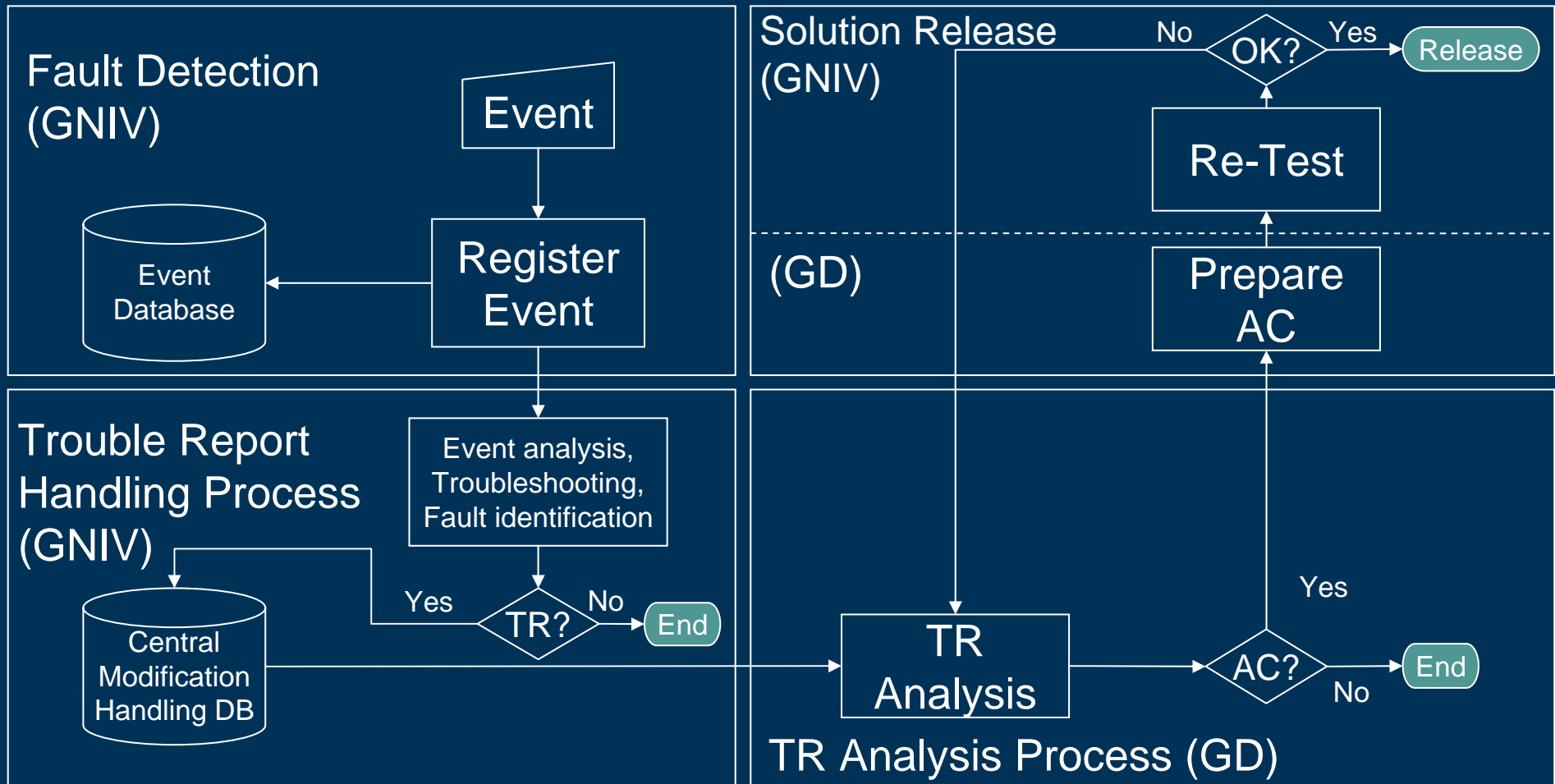


# Define



- Goal : Increase productivity in order to keep a flat budget, while increasing operative margins.
  - ➔ Reduce cost level for verification and maintenance operations in order to
  - ➔ Increase productivity

# Fault Handling Process



# Measure

TR ID	task
1	Screening
2	Check if TR is written
3	Check if problem is known
4	Check if TR description is complete
5	Contact TR issuer for more info
6	All other screening related activities
7	Assignment to handle
8	Analysis
9	Get expert support
10	Check if TR description is complete
11	Contact TR issuer for more info
12	All other analysis related activities
13	AC preparation
14	Check if an AC is needed
15	Check if working on a canceled AC
16	Prepare PAC
17	Check for mapping TRs
18	1. DC
19	2. DC (if applicable)
20	Update AC
21	Finalize TI
22	Check if AC test is possible at DM
23	AC Test
24	Final DC
25	All other AC related activities
26	Prepare TRA
27	Final check
28	Test load
29	TR closure activities
30	
31	
32	
33	
34	
35	

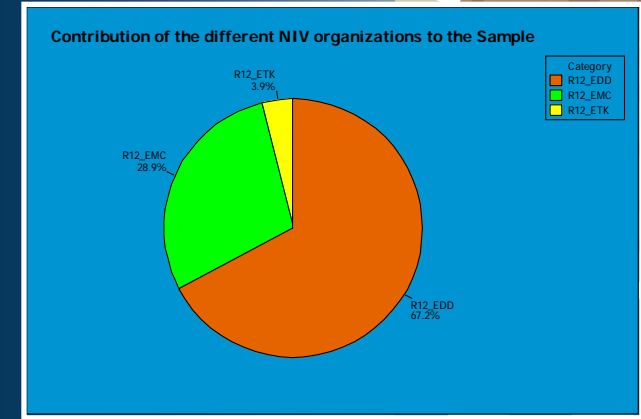
Measurement Protocols  
(450 measurements)

## Additional Information

(900 records)



## Representativeness



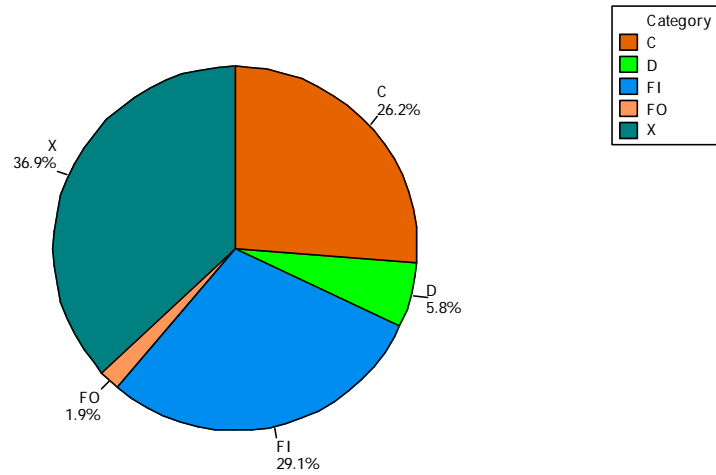
Measurement System Analysis

Attribute Agreement Analysis:  
72% agreement before, 97% after optimization

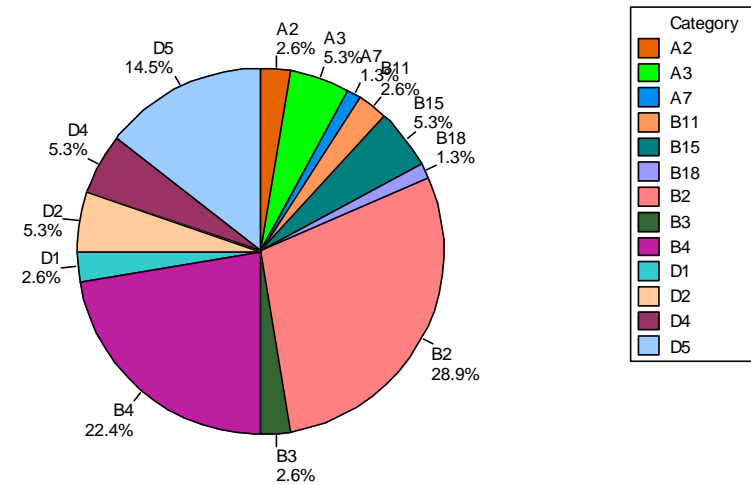
# Analyze

**Distribution of final states for faults**

Data cut on  $t \leq 1d$  and  $cost \leq 1mh$



**Answer Code Distribution for TRs**



Worksheet: NIV Analysis; 1/18/2006; EDD/XS/P Dirk Macke

- 30% of fault handling effort in verification does not lead to a trouble report; these are cost for troubleshooting tools, test environment etc.
- 25% of all trouble reports are turned down due to insufficient information, misunderstanding of functionality, or inability to reproduce the fault

# Improve

- Perceptions and assumptions could be replaced by solid figures.
- Some assumptions failed the reality check: This was a great enabler, to create acceptance of the other results as well.

## Improvement Areas:

1. Reduce the amount of trouble reports, which are turned down due to insufficient input quality
2. Reduce documentation overhead and internal troubleshooting
3. Improve productivity

This was achieved by

- Introduction & establishment of a fault handling process in GNIV
- Improved communication by introduction of a trouble report skeleton
- Monitoring of GD maintenance initial response time



# Control

- The GNIV fault handling process has been handed over to the process owner(s) after implementation (local line organizations).
- Peer reviews with senior and master engineers have been introduced to ensure usage of the trouble report skeleton, and to improve trouble report quality.
- The monitoring of initial response time from maintenance has been handed over to the process owner (GD Project Office).
- Key Performance Indicators (KPIs) have been defined for all vital process steps, and are monitored on a weekly basis to enable statistical process control.

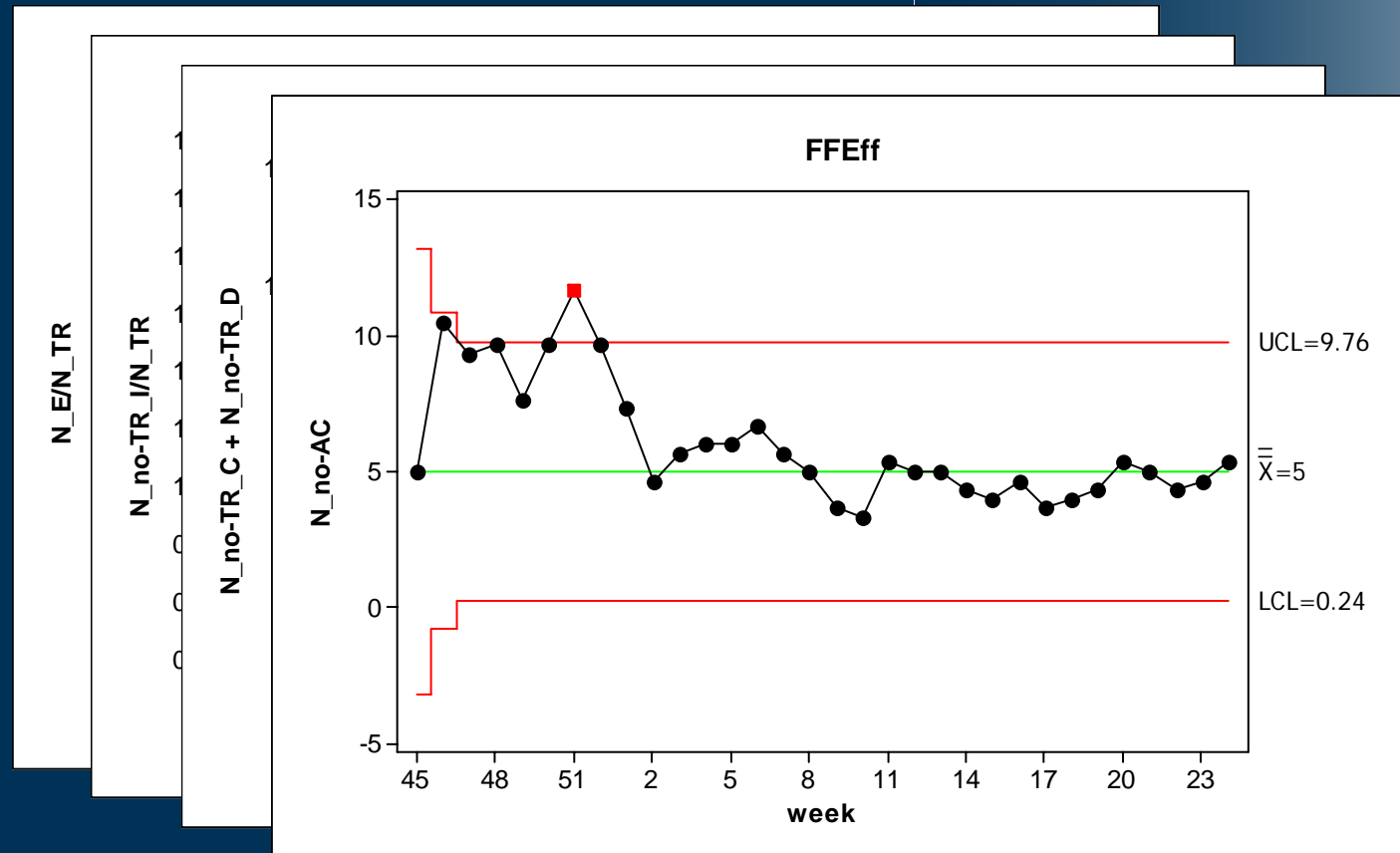
# Control – Key Performance Indicators

$$FHE = \frac{N_E}{N_{TR}} \quad FFE = \frac{N_{no-TR\_I}}{N_{TR}}$$

$$FHEff = N_{no-AC} \quad FHEff = N_{no-TR\_C} + N_{no-TR\_D}$$

- Four KPIs have been identified in order to measure efficiency & effectiveness of the fault handling process, as well as process adherence:
  - Fault Handling Efficiency ( $FHE$ ,  $N_E$  = number of events found,  $N_{TR}$  = number of TRs written) → The smaller  $FHE$ , the less time and money is spent on issues not leading to a TR (and by this to a product improvement).
  - Fault Finding Efficiency ( $FFE$ ,  $N_{no-TR\_I}$  = internal GNIV faults, not leading to a TR) → The smaller  $FFE$ , the less time and money is spent on issues not related to product quality in the first place.
  - Fault Handling Effectiveness ( $FHEff$ ,  $N_{no-TR\_C}$  = issues, which have been logged but do not constitute a fault/issue at all and  $N_{no-TR\_D}$  = duplicated issues, which have been logged twice, but belong to one and the same fault) → The smaller  $FHEff$ , the less time and money is spent on 'assumed' faults.
  - Fault Finding Effectiveness ( $FFEff$ , the number of faults handled in GNIV, but not resulting in a correction in GD ( $N_{no-AC}$ ), and by this not improving product quality → This is a direct measurement of COPQ.

# Results - Statistical Process Control



- Red lines: Control limits
- Green line: Target Level to be reached in order to realize the projected savings/cost improvements
- Start of implementation: wk48-50 (depending on site/country)

**ERICSSON** 

**TAKING YOU FORWARD**