



Houston, we have a problem!

Pierre Almén, ImproveIT
Fact-Based IT Improvements
pierrea@coolmail.se
2015-11-18



Houston, we have a problem!

■ Background

- Case 1 – Background & Scope
- Case 1 – Create metrics
- Case 1 – Result
- Case 2 – Background & Scope
- Case 2 – Findings & Result



Background

Pierre Almén - ImproveIT

- FP usage from 1984
- CFPS – Certified Function Points Specialist 1994
- CSMS Gold Level – Certified Software Measurement Specialist 2006
- IFPUG Committee member from 2008 and Board member from 2014
- Reviewer and co-author of "The IFPUG Guide to IT and Software Measurement"
- Responsible for a FP / Software Metrics network in Sweden (1990-)
- Performance improvement and sourcing studies of systems development and maintenance at major Nordic companies and organisations
- Developer / project leader / systems development & maintenance manager within IBM



Background

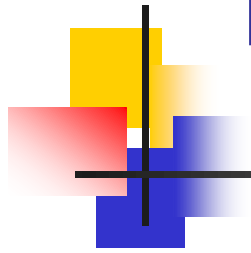
Systems development & maintenance often 50% of IT budget

Standish Group CHAOS Report 2014

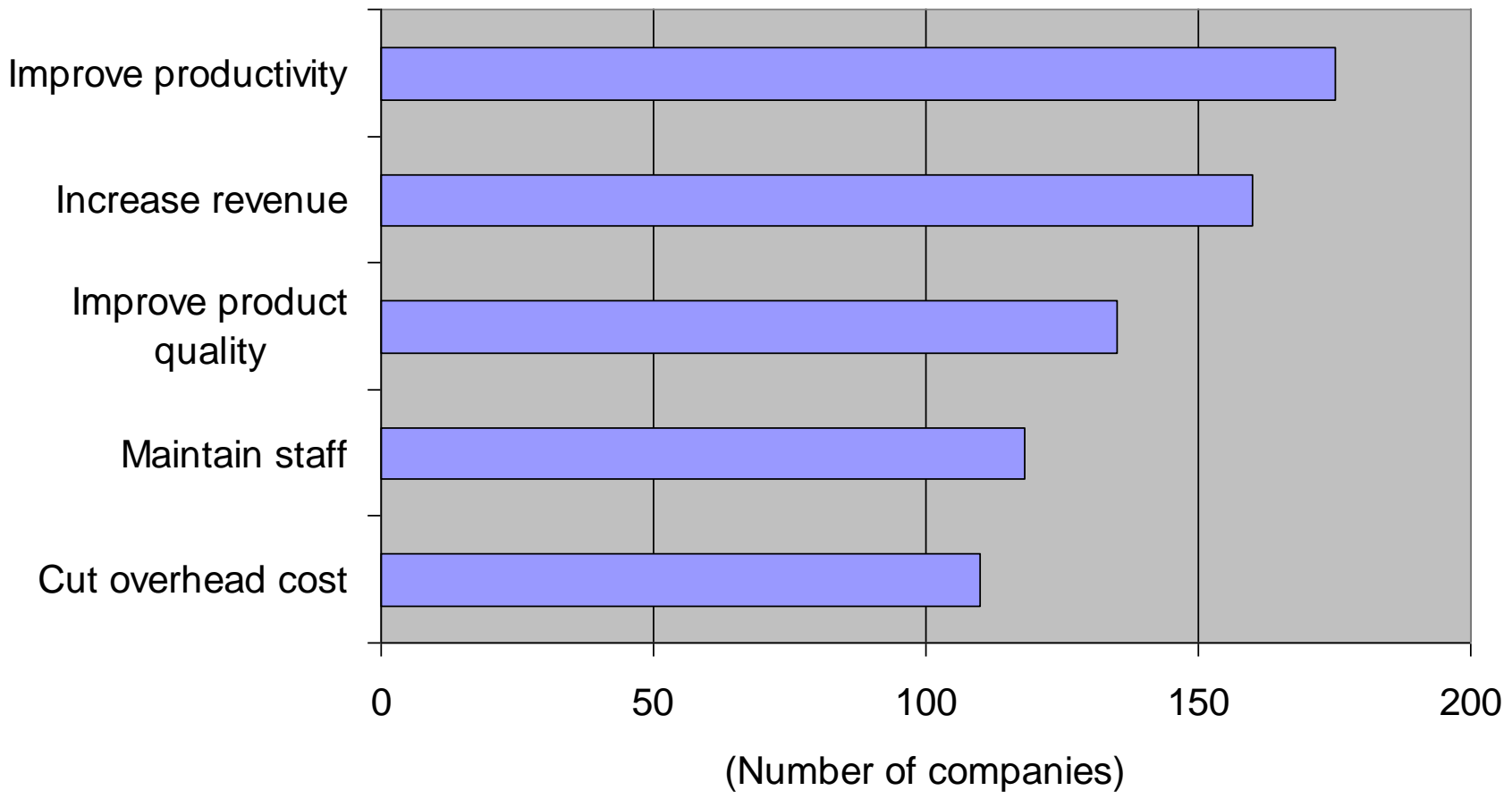
- Successful projects 16 % (32% 2009)
 - Delivered on time, on budget, with required features and functions
- Challenged projects 53 % (44% 2009)
 - Are late, over budget, and/or with less than the required features and functions
- Failed projects 31 % (24% 2009)
 - Cancelled prior to completion or delivered and never used



Background

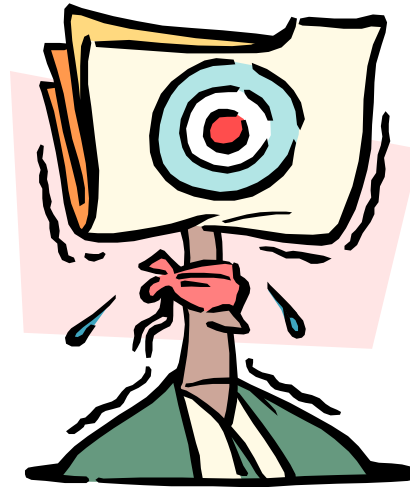


Top 5 Business Challenges





Background



"I'm making a decision! Stop confusing me with facts!"

"Without metrics, you're just another person with a different opinion"



Houston, we have a problem!

- Background
- Case 1 – Background & Scope
- Case 1 – Create metrics
- Case 1 – Result
- Case 2 – Background & Scope
- Case 2 – Findings & Result



Client Case 1

Case background

- Supplier shall develop and deliver a system including major business processes (mini-ERP)
- Quality criterias exists in the contract and the supplier is unsure if they are market-like and if they can be fulfilled
- Primary purpose was to do a market comparison of relevant quality metrics
- Secondary purpose was to prognosticate the upcoming resources needed for application maintenance



Scope

- Create metrics for the project
 - Based on Function Points (FP)
 - Estimate # of system faults / FP
 - Estimate # of acceptance test faults / FP
 - Estimate project productivity (FP / Person month)
- Find metrics in benchmarking DB for similar projects and following metrics for similar applications
 - Reported faults / 1000 FP
 - Faults distribution in three categories : "Critical, Major, Other"
 - Productivity for application maintenance (user support, faults fixing and non-functional enhancement) : # of FTE / 10000 FP
- Estimation of Function Points
 - Estimation from LOCs
 - Estimation from user functions



Houston, we have a problem!

- Background
- Case 1 – Background & Scope
- Case 1 – Create metrics
- Case 1 – Result
- Case 2 – Background & Scope
- Case 2 – Findings & Result



Creation of metrics

Estimation of project / application functional size

- Convert lines of code (LOC) to Function Points using converting factors from different sources
 - Gives probably bad result for a single project / application, can give better result for a whole portfolio
 - Result based on data from several suppliers of conversions tables:

12700 – 24000 FP ?





Creation of metrics

Estimation of project / application functional size

- Convert Use Case Points to Function Points
 - Uncertain basic data existed for Use Case Points
 - Counting was done in early stage, many functions may be missing
 - No standardized way of documenting Use Cases
 - Limited conversion data existed:

4050 FP ?





FP sizing

Estimation of project / application functional size

- Fast Function Point counting of first sub system (estimated by the supplier to be just over 13%)
- Counting transaction FPs, data FPs estimated
- Assuming just over average complexity for transaction FPs
- Added 10% for missing functionality

4900 FP



FP sizing

Estimation of project / application functional size

- Fast Function Point counting of second sub system (estimated by the supplier to be approx 20%)
- Counting transaction FPs, data FPs estimated
- Assuming just over average complexity for transaction FPs
- Added 10% for missing functionality

8150 FP



FP sizing

Estimation of project / application functional size

- Summarizing of both sub systems gives a total FP size of

6850 FP

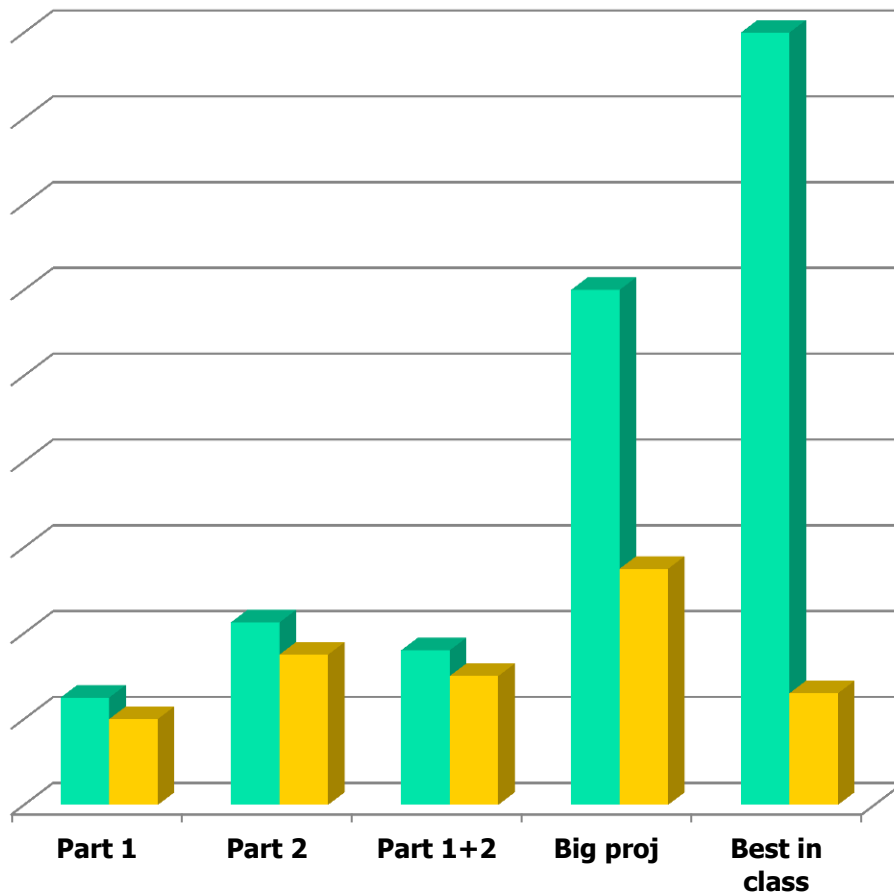


Houston, we have a problem!

- Background
- Case 1 – Background & Scope
- Case 1 – Create metrics
- Case 1 – Result
- Case 2 – Background & Scope
- Case 2 – Findings & Result



Result 1(5)

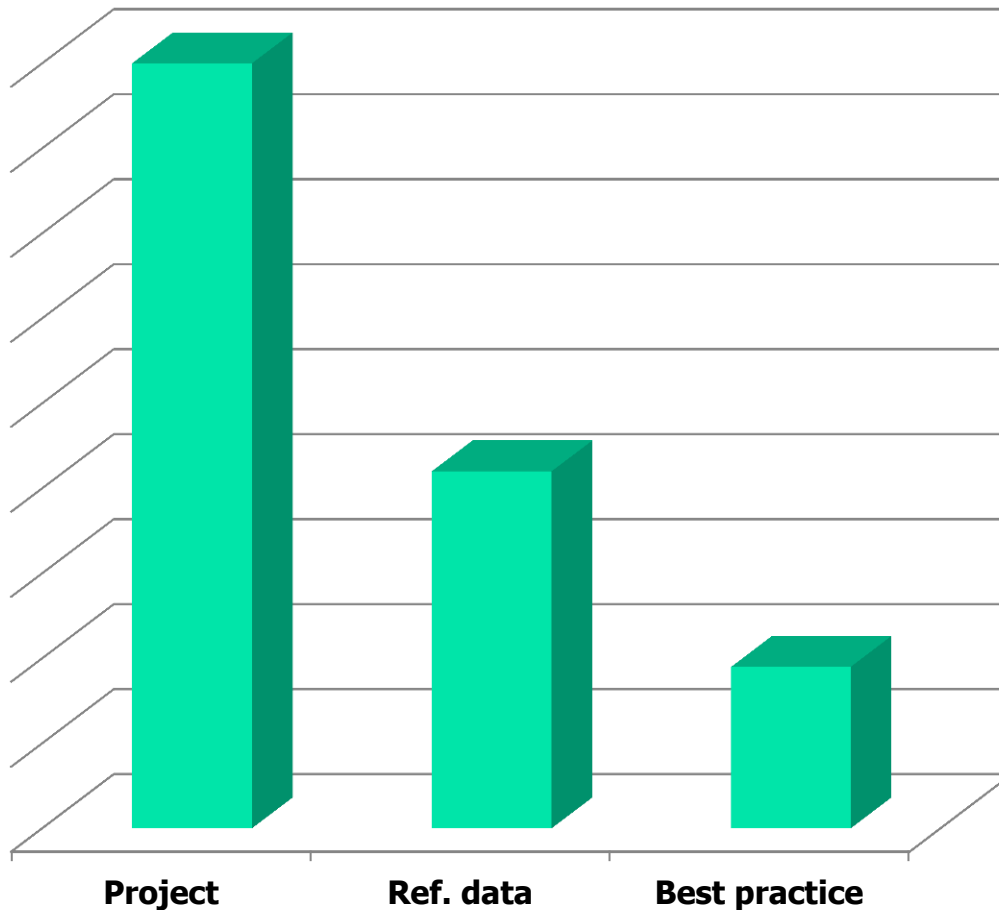


Project showed lower productivity and delivery capacity compared to reference data

- Productivity
- Del. Capacity



Result 2(5)

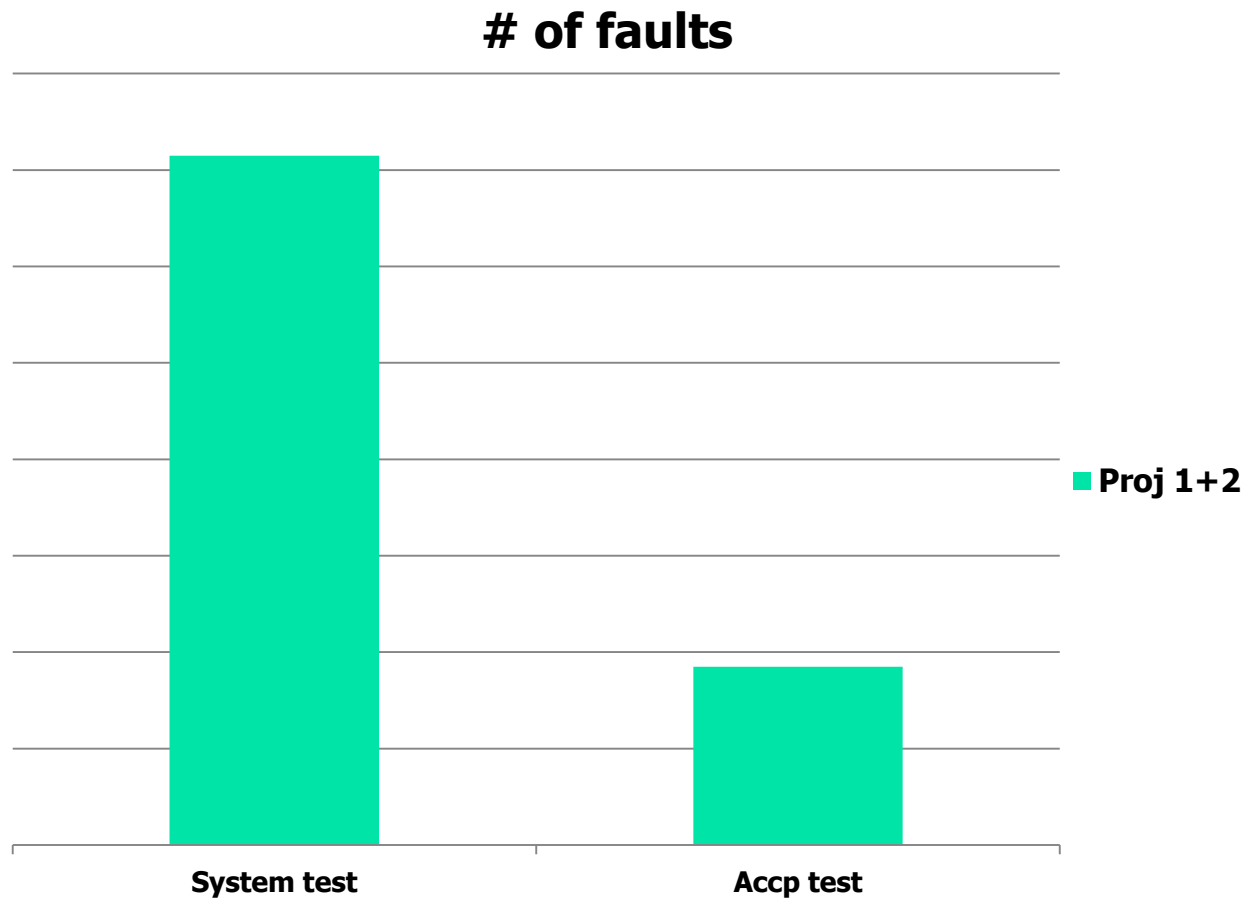


The project has had higher staffing (more reported hours per day) vs reference data

■ Hrs/day

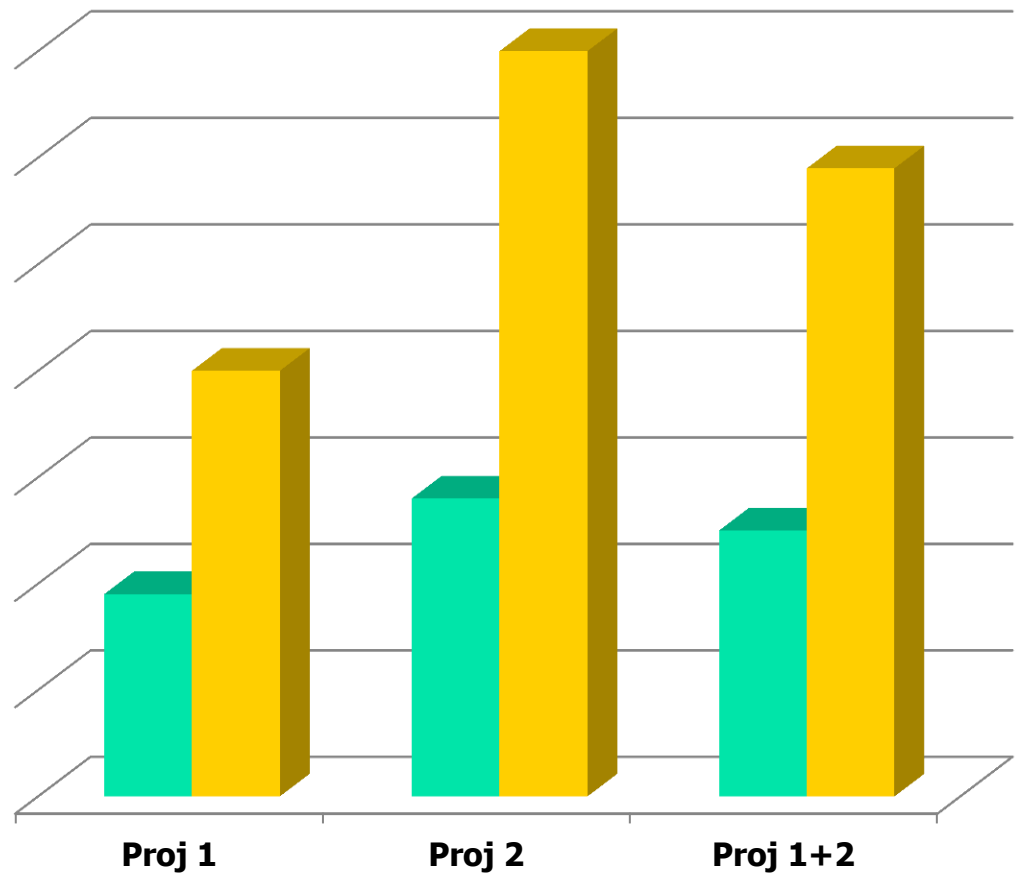


Result 3(5)





Result 4(5)



■ # FTE
■ # Faults

Based on reference data the number of FTEs to maintain the application and # of faults in production are prognosticated.



Result 5(5)

Appl faults



Distribution of faults for big appl.



Houston, we have a problem!

- Background
- Case 1 – Background & Scope
- Case 1 – Create metrics
- Case 1 – Result
- Case 2 – Background & Scope
- Case 2 – Findings & Result



Client Case 2

Case background

- Part of application portfolio outsourced to offshore supplier
- Penalty if supplier fails to meet metrics in the contract
 - Resolution time for defects
 - Defect reduction
- Monthly report from supplier showed green metrics
- Complaints at manager meetings about time to fix defects
- Defect reduction metric showed fewer defects than expected



Scope

- Investigate why there are different opinions by
 - Going through the contract
 - Collecting reports and analyze the data
 - Interviewing both people from client and supplier



Houston, we have a problem!

- Background
- Case 1 – Background & Scope
- Case 1 – Create metrics
- Case 1 – Result
- Case 2 – Background & Scope
- Case 2 – Findings & Result



Findings

- Defect resolution time PI had four levels based on defect severity
- Calculation of resolution time done based on
 - When supplier was notified
 - When a temporary workaround solution or permanent fix existed and was accepted by customer
 - Three levels measured in hours and fourth in days



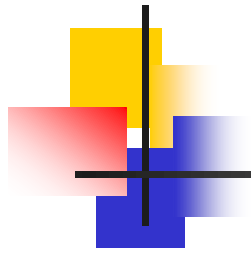
Findings

- Further investigation found that
 - Client application responsible was not aware of contract details
 - Hours was only counted within "business hours"
 - In reality, resolution time for high level defects about same as for lowest level
 - Hours was only counted when defect case was assigned to supplier
 - Supplier sent queries to client and stopped the clock (often a ping-pong match!)
 - Supplier had not access to production DB and often sent DB queries in mail and stopped the clock
 - For the defect reduction PI, the supplier excluded all defects where the clock was stopped



Result

- The reports were updated to show the PI's from a business point of view
 - Report now includes total resolution time and total defect backlog regardless of who the case is assigned to



?



Obrigado!

Thanks!

Tack!