Silver Bullets

Do they hit the mark?

a look at the impact of tools & techniques used on IT projects

Peter R Hill – CEO

International Software Benchmarking Standards Group

www.isbsg.org

What we will cover

- The ISBSG Development & Enhancement Repository
- The tools & techniques that have been used on projects in the ISBSG repository
- The impact that these tools & techniques appear to have on the projects
- What works and what doesn’t

The Search for a Silver Bullet

The Lone Ranger

Silver Bullets - Past Pretenders?

- CASE tools?
- 4GL languages?
- Code generators?
- Object Oriented techniques?
- Re-use?
- Others?

The ISBSG Repository

- The ISBSG Development/Enhancement repository has data on 4,150 completed projects
- Not all projects contain data about tools and techniques used
- A sub-set of the Repository can be licensed on a CD or all the data via a corporate subscription.

Definitions

- ISBSG = International Software Benchmarking Standards Group
- FSM = Functional Size Measurement
- FP = Function Point
- PDR = Project Delivery Rate

  = Hours per functional size unit (eg: hours per function point)
- Productivity = Productivity of development team only (L1).
- PWE = Project Work Effort

Techniques

Under the heading of “Techniques” we include:

- Methodologies
- Standards
- Prototyping
- Rapid Application Development
- Development Techniques
- Modelling (Data, Process, Event)
- Object Oriented Analysis & Design

Tools

Under the heading of “Tools” we include:

- CASE (various levels)
- Project Management
- User Requirements & Specification
- Performance Monitoring
- Testing
- Debugging

Techniques - the trade-off

- In general terms Techniques tend to be used to improve the “quality” and delivery of systems.
- Productivity may be affected.

(You need to know this when you are estimating or benchmarking).

Methodologies

- Productivity: Median PDR in hrs per fp - 12.2hrs with a methodology, 7.6hrs without.
- Speed of Delivery: little difference, 32 fp pp per month.
- Quality: Defects per 1000fp – 3 with a methodology, 28 without.
- Team Size: Slightly larger with a methodology, 6 rather than 5.
- Testing: 23% of effort with a methodology, 19% without.

Methodologies - summary

Although methodologies do add overhead, and as a consequence add effort, they appear to make a significant difference to quality, with a much lower occurrence of defects.

CMM & CMMI

- Productivity: Median PDR in hrs per fp – 8.9hrs with CMM(I), 10.2hrs without.
- Speed of Delivery: 3.6 fp pp per month with CMM(I), 5.8 without.
- Quality: Defects per 1000fp zero with CMM(I), 3.3 without.
- Team Size: Larger with CMM(I), 11.5 rather than 6.

CMM & CMMI Summary

- clearly and significantly associated with larger project teams.
- associated with increased duration.
- associated with fewer total defects being delivered.
- speed of delivery is slower.

ISO Standards

- Productivity: Median PDR in hrs per fp - 8.1hrs with ISO stds,10.3hrs without.
- Speed of Delivery: 3.8 fp pp pm with ISO, 5.7 without.
- Quality: Insufficient data
- Team Size: Larger with ISO stds, 13 rather than 6.

Standards Summary

Standards are significantly associated with:

- Larger projects.
- Longer duration.
- Larger project teams.
- Fewer FPs delivered per team member per month.
- Fewer delivered defects and lower defect density.

Prototyping

- Productivity: Median PDR in hrs per fp – 7.5hrs with Prototyping,10.9hrs without.
- Speed of Delivery – 8.6 fp pp per month with prototyping, 5 without.
- Quality: *Insufficient data*
- Team Size: No difference – median of 6
- Projects using prototyping spend more time in the Build phase.

Rapid Application Development

- Productivity: Median PDR in hrs per fp – 7.7hrs per fp with RAD,10.4hrs without.
- Speed of Delivery: 8.6 fp pp per month with RAD, 5.3 without.
- Quality: *Insufficient data*
- Team Size: Slightly smaller with RAD, 6 rather than 7.

Development Techniques

- Productivity: Median PDR in hrs per fp – ~8.5hrs for classical + other and for non-classical only. Classical only is 14.3hrs.
- Speed of Delivery: little difference ~5 fp pp per month with classical methods, or non-classical.
- Quality: Insufficient data.
- Team Size: Tends to be bigger for non-classical - 8 people rather than 6.

Joint Application Development

- Productivity: Slightly better with JAD – 9.3 compared to 10.2 for non-JAD.
- Speed of Delivery: Slightly better with JAD – 6.9 fp pp pm rather than 5.4.
- Quality: *Insufficient data*
- Team Size: No difference – median team size of 6 with or without JAD.

Object Oriented Analysis

- Productivity: Lower using OO analysis 12.1 hrs per fp compared to 10 for non OO analysis.
- Speed of Delivery: Lower with OO – 4.0 fp pp pm rather than 5.6.
- Quality: *Insufficient data*
- Team Size: Bigger team sizes with OO – median team size of 9 with, 6 without.

Summary for techniques

Methodologies are associated with lower defect rates but increased effort and a slower speed of delivery.

CMM(I) & Standards are associated with larger teams, increased duration, lower defects, slower speed of delivery.

Prototyping, RAD & JAD are associated with positive project performance for effort and speed of delivery, but there is insufficient data to judge if there is an impact of defect rates.

TOOLS

Requirements/Specification Tools

- Productivity: Median PDR in hrs per fp – 10.6hrs with the tools,16.3hrs without.
- Speed of Delivery: 5.2 fp pp per month with the tools, 2.7 without.
- Quality: *Insufficient data*
- Team Size: Slightly higher with the tools, a median of 8 rather than 7.

Project Management Tools

- Productivity: Median PDR in hrs per fp – 11.7hrs with the tools,5.2hrs without.
- Speed of Delivery: *Insufficient data.*
- Quality: Better with these tools - 10 defects per 1000fp without tools, only 5 with.
- Team Size: Significantly higher with the tools, median of 7 rather than 3.

CASE Tools

- Productivity: Median PDR in hrs per fp – no difference ~9.
- Speed of Delivery: Slightly better without CASE: 7.9fp pp pm compared to 7.
- Quality: 11 defects per 1000fp without tools, 1 with.
- Team Size: Bigger teams using tools, 7 rather than 5.

Other CASE Tool Observations

- Effort is considerably lower on projects using CASE tools with code generation.
- CASE tools (of all sorts) have a significant impact in reducing defect density.
- Upper CASE tools are associated with greater Specification effort.

So What Works?

- CMMI & Standards – Lower defect rates
- Prototyping (Seems to improve PDR)
- Rapid Application Development (PDR and Speed of Delivery)
- Joint Application Development (PDR and Speed of Delivery)
- Case Tools – lower defect rates

Use of Tools & Techniques

- Methodologies are the only technique used by a large majority of projects in the ISBSG repository.
- CASE tools, Testing tools and Data Modelling have moderate usage but all other tools & techniques have very low usage.

Is there a Silver Bullet?

It appears not!

But there are useful tools & techniques

Questions?