Developers Want Requirements, but Their Project Manager Doesn't; and A Possibly Transcendent Hawthorne Effect

Daniel Isaacs and Daniel M. Berry
University of Waterloo

Outline

- **Introduction**
- **Related Work**
- **Case Study Description**
- **Case Study Findings**
- **Discussion**
- **Conclusions**

Acronyms

RE	**requirements engineering**
SW	**software**
SDLC	**SW development lifecycle**
CS	**case study**
FA	**first author, Daniel Isaacs**

Introduction

A common observation or complaint:

Despite the demonstrated benefits of a serious RE process on the containing SDLC, …

too many SW construction projects skip RE and go directly on to development, …

because of pressure from above over a short deadline for completion of the SW.

This Talk Presents

This talk presents a CS of a project that went straight to the development phase, understanding only few requirements, …

a CS of one project by one organization to integrate into its suite of products a product from another organization that the first organization acquired.

Conflation

This talk conflates “requirements” with “requirements specification” …

as did the members of the project examined in this CS.

They used the terms interchangeably, and …

“lacking requirements” means “lacking both requirements and a requirements specification”.

Goal of the CS

The goal of the CS was to provide some insight on how the lack of RE in a SW development project affected the quality of the developed SW product.

The CS used a questionnaire filled in by all the project's members except the FA.

The questionnaire asked project members for their views on various RE topics.

Related CS Work

The role of RE has been described in literature as to:

- **identify the problem,**
- **identify the stakeholders,**
- **explore different solutions with the stakeholders, and**
- **decide what to implement.**

Related CS Work

- **Hofmann & Lehner**
- **Workshop on RE Payoff**
- **Panel on RE Payoff at RE'05**
- **Damian *et al***
  We used their questionnaire!

Hofmann & Lehner

Hofmann and Lehner (2001)

- **conducted surveys and interviews of 76 stakeholders (managers, requirements analysts, customers, and quality assurers)**
- **to study 15 RE teams in 9 development organizations in telecommunications and banking.**

Hofmann & Lehner, Cont'd

Among their conclusions:

- **The most successful teams expended an effort on RE that was twice that expended by the least successful teams.**

- **The most successful teams did RE for a greater portion of their SDLCs than did the least successful teams.**

Workshop on RE Payoff

Full-day workshop in 2005 on the theme: upfront RE pays off in an improved development, software, or both.

Each presentation described a CS of the development of real-life or substantial research software in which thorough RE was done before development began.

Panel at RE'05

A 1.5-hour summary of this workshop was presented at RE'05 in a panel titled "To do or not to do: If the RE payoff is so good, why aren't more companies doing it?"

Damian et al.

Damian et al. (2004, 2005, 2006) report the results of a 30-month, 3-stage, explanatory CS, using questionnaires, interviews, and document inspection, of the RE process at the Australian Centre for Unisys Software (ACUS).

Damian et al., Cont'd

During the study, ACUS was actively improvings its RE process, following a CMM assessment.

The study concluded that the RE process improvement did indeed improve ACUS's downstream software development and the resulting software.

CS Context

The CS of RE practices was conducted at Ontario office of a Canadian company X.

X founded in 1981, and has about 600 employees in 26 offices around the world.

About 225 – 250 are in the Ontario office.

X sells about 20 product suites and 100 bundleable services.

US$99.2M revenues in fiscal 2011.

Ontario Office Characteristics

From now on,
“X” means “the whole of X” and
“O” means “Ontario office of X”.

Average SW development project at O

- **took 18 months, with range 12 – 24 months,**
- **generated 2.8MB of delivered source code, and**
- **followed a so-called agile SDLC.**

Agility?

Agility was very lightweight & in name only.

O did not carefully follow all steps (e.g., continuous customer presence), …

that ensure wide distribution of requirements knowledge in absence of requirements documentation.

Authors' Opinion about O

As in many places, O's "Agile" was a fancy name for doing the old-fashioned seat-of-the pants SDLC, …

with no RE, no documentation at any time, skipping everything that wastes time, and with a lot of scrambling near and after the deadline …

and then a lot of extra work to fix the mistakes.

Authors' Opinion, Cont'd

A relevant quotation:

"There's never enough time to do it right, but …

there's always enough time to do it again."

An X Acquisition

X acquired another Canadian company, Y.

Y's main product is PY.

X acquired Y mainly to incorporate PY's functionality into its own products.

O Project to Build PX

O began a project to build PX in March 2008.

From now on, this project is called “the project”.

Project scheduled for 18 months; required 24.

Client was representative of first X customer that agreed to beta test new PX.

Staffing for the Project

Project started with team of 16.

But, 8 quit in first 9 months from job dissatisfaction, …

leaving 8, including the FA, at the time of the CS.

These 8 included 7 developers, including the FA, and 1 quality assurer.

The Non-Royal “We”

The FA was a member of the project that was studied.

Therefore, each of “we”, “us”, and “our” includes the FA, …

and does not include the second author.

Challenges in the Project

One significant challenge we faced when we started the project was our lack of knowledge of PY's domain.

PY's developers and other stakeholders, such as end users, were geographically separated from the PX project team.

When Y became part of X, all PY developers, who had domain knowledge about PY, quit rather than become X employees.

Initial Conditions

- **X's senior management communicated to PX developers in O that their job was to replicate the functionality of PY.**

  No more, no less functionality than PY had.

- **PY's functionality had to be migrated to a different technology, in order to incorporate the functionality into O's suite of SW.**

Initial Conditions, Cont'd

- **The project manager at O communicated PX's requirements as a one-sentence requirements specification: …**

  "Mimic this Webpage." while pointing to the Webpage implemented by PY.

Initial Conditions, Cont'd

- **PY's functionality was not defined or documented anywhere.**

  Information sufficient for a smooth development was not provided.

  Thus, the developers did not fully understand what was required to build PY.

Initial Conditions, Cont'd

- **The implementation of PX relied heavily on each developer's own interpretation, …**

  a serious problem since each developer's interpretation was different from those of the others.

Motivation for CS

There is evidence that RE benefits the subsequent SW development.

However, in many industrial organizations, RE is completely ignored.

Therefore, the FA wanted to conduct a CS of the impact of missing RE in a SW development product.

Data Collection Procedures

The FA invited all 7 other team members to participate in a questionnaire.

All 7 returned filled in questionnaires, for a 100% response rate.

The questionnaire was based on the one used by Damian et al (2005), with some changes and some additions.

Form of the Questionnaire

Some questions asked for an answer in a 5-point Likert scale:

Very Z, Z, Neutral, ¬Z, Very ¬Z

Some questions asked for a short answer in the respondent's own words.

Some questions asked for both with the Likert scale answer to be chosen first.

Limitations

The FA's presence on project team → possible bias.

But the FA tried to mitigate bias by asking follow up questions.

The second author's 46 years' experience moderated the bias.

In any case, no attempt to generalize beyond the CS.

Questions in Questionnaire

Q1 How important do you feel requirements are? (LS)

Q2 How do you feel that the lack of requirements influenced your work? (OE)

Q3 Based on your experience on this project, would you spend more or less time in the requirements phase of the development? (LS) Why? (OE)

Questions, Cont'd

Q4 In your design, coding, testing, or documentation activities, how important was it to understand the features and technical requirements? (4×LS)

Q5 In contrast to previous experiences, has there been more or less rework during development (but before deployment)? (LS)

Q6 How do you believe the lack of requirements improved or deteriorated: (a) productivity and (b) product quality. (OE)

Questions, Cont'd

Q7 Did the lack of requirements affect your cost estimation process? (LS)

Q8 When thinking about your estimates, if any, can you think of reasons for discrepancies? With respect to (a) Design, (b) Implementation, (c) Testing and (d) Documentation. (OE)

Q9 How could you have improved your estimations? (OE)

Questions in Questionnaire

Section A: General feedback about the lack of requirements

Q1. How important do you feel requirements are?

A:Far More B:More C:About the same D:Less E:Far Less

Q1's scale is admittedly wrong because, e.g., it leaves unanswered "Far More than What?"

Questions, Cont'd

The scale should have been:

A:Very Important B:Important C:Indeterminate
D:Somewhat important E:Not important at all

But FA's talking with respondents made it clear that they answered as if the scale were correct!

Questions, Cont'd

Q2. How do you feel that the lack of requirements influenced your work?

Q3. Based on your experience on this project, would you spend more or less time in the requirements phase of the development? Why?

A:Far More B:More C:About the same D:Less E:Far Less

Questions, Cont'd

Section B: Definition of the problem, and the possible benefits from understanding the problem

Q4. In your design, coding, testing, or documentation activities, how important was it to understand the features and technical requirements?

A:Very Important B:Important C:Indeterminate
D:Somewhat important E:Not important at all

Questions, Cont'd

Q5. In contrast to previous experiences, has there been more or less rework during development (but before deployment)?

A:Far More B:More C:About the same D:Less E:Far Less

Q6. How do you believe the lack of requirements improved or deteriorated: (a) productivity and (b) product quality.

Questions, Cont'd

Section C: Estimation

Q7. Did the lack of requirements affect your cost estimation process?

A:Strongly Agree B:Agree C:No Effect D:Disagree E:Strongly Disagree

Questions, Cont'd

Q8. When thinking about your estimates, if any, can you think of reasons for discrepancies? With respect to (a) Design, (b) Implementation, (c) Testing and (d) Documentation.

Q9. How could you have improved your estimations?

Principal CS Findings

1. **6 of the 7 respondents (5 developers, 1 quality assurer) wanted more to much more requirements and considered them important to very important. (More about 1 outlier later)**
2. **Lack of requirements hindered the project in many ways and caused lots of rework.**
3. **The PX project was not different from other, previous projects at O.**

CS Findings

These are discussed by Section:

- ***Section A is about general feedback about the lack of requirements.***
- **Section B is about the definition of the SW development problem and the possible benefits from understanding the problem.**
- **Section C is about estimating the project's effort, cost, and time for development.**

Q1

How important do you feel requirements are?

Answers showed that …

even though requirements were largely missing in the project, all seven project members thought that requirements were at least important, while five of these seven, thought that requirements are far more important.

Q2

How do you feel that the lack of requirements influenced your work?

Answers showed that the lack of RE influenced their work in many ways.

Q2, Cont'd

In project members' own words:

"It would have been easier to put all of the components together."

"… too much guessing regarding expected results."

"… not enough time to develop good test cases, which made testing not being complete enough."

Q2, Cont'd

“It hampered my work creating unexpected results.”

“We had to go back and rewrite sections of the code, causing project timelines to increase.”

Q3

Based on your experience on this project, would you spend more or less time in the requirements phase of the development?

Answers showed that …

six of the seven project members thought that they should have spent more time on the requirements phase, with two of these six saying “far more” instead of just “more”.

Q3, Cont'd

Only one developer of the seven project members thought that he should have spent far less time on the requirements phase.

Q3, Cont'd

In project members' own words:

"RE provides a better understanding of what exactly needs to be done."

"RE gives a crystal clear understanding of what the user is looking for and what he/she wants to achieve from this project."

Q3, Cont'd

"RE would have prevented many different problems and miscommunications with management."

"I would still need to spend a lot of time in the designing phase."

CS Findings, Cont'd

These are discussed by Section:

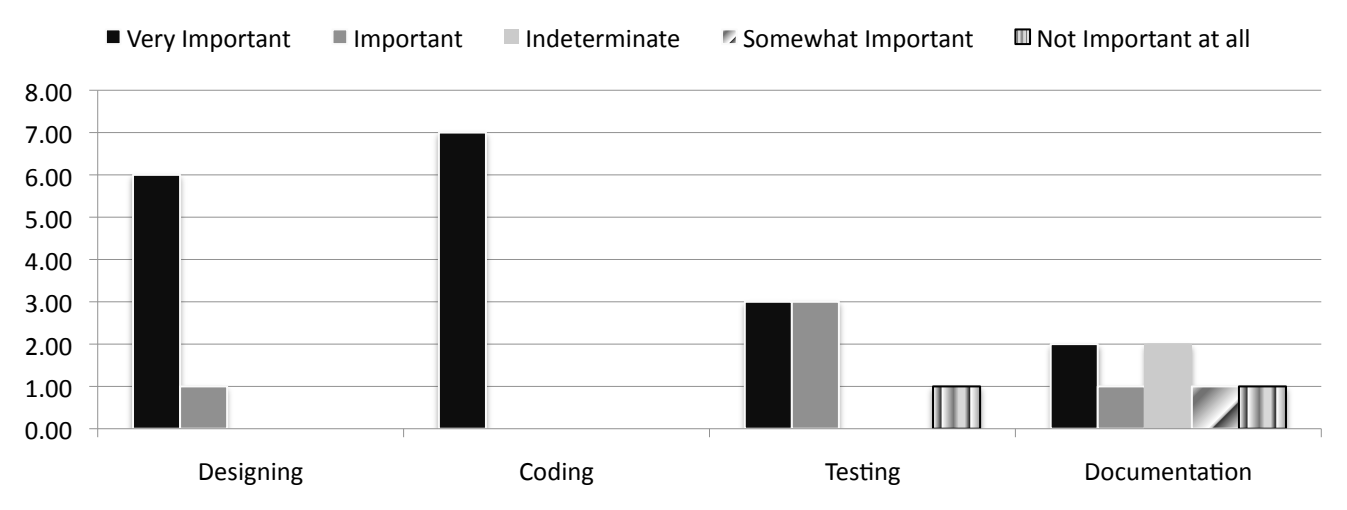
- **Section A is about general feedback about the lack of requirements.**
- ***Section B is about the definition of the SW development problem and the possible benefits from understanding the problem.***
- **Section C is about estimating the project's effort, cost, and time for development.**

Q4

How important was to understand the features and technical requirements?

Answers to Q4

in the graph on the next slide


Very Important
Important
Indeterminate
Somewhat Important
Not Important at all
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
Designing
Coding
Testing
Documentation

Q5

In contrast to previous experiences, has there been more or less rework during development (but before deployment)?

Answers showed that …

six out of seven project members thought the amount of rework done in the current project was about the same or more than in previous projects.

Q5, Cont'd

Three of these six project members indicated there was more work.

Only one project member thought there was far less rework than in previous projects.

Q6

How do you believe the lack of requirements improved or deteriorated: *(a) productivity* and (b) product quality?

In project members' own words:

"Productivity was negatively affected by the amount of work that had to be redone."

Q6, Cont'd

"Lack of RE negatively impacted productivity."

"Lack of information always results in unexpected results."

Q6

How do you believe the lack of requirements improved or deteriorated: (a) productivity and *(b) product quality*?

In project members' own words:

"It deteriorated product quality because you have to keep changing the design and code as the requirements change."

Q6, Cont'd

"It reduced the initial product quality, however the later product is still of good quality, but it took much longer to get to this point due to the lack of proper requirement."

CS Findings, Cont'd

These are discussed by Section:

- **Section A is about general feedback about the lack of requirements.**
- **Section B is about the definition of the SW development problem and the possible benefits from understanding the problem.**
- ***Section C is about estimating the project's effort, cost, and time for development.***

Q7

Did the lack of requirements affect your cost estimation process?

Answers showed that …

the lack of RE affected the cost estimation process negatively.

In particular, five of the seven project members strongly agree that the lack of RE affected their estimation, while two said that they agree that it made a difference.

Q8

When thinking about your estimates, if any, can you think of reasons for discrepancies? With respect to (a) Design, (b) Implementation, (c) Testing and (d) Documentation.

In project members' own words:

"The design had to be reworked several times due to new discoveries and additional requirements."

Q8, Cont'd

"It [the design] was usually a set back since there was no proper understanding of what the system was required to do."

"The implementation had to be restarted due to the changes in design."

"It [the documentation] was very late as it had to be started late in the project."

Q9

How could you have improved your estimations?

In project members' own words:

"Have a list of requirements and a design document about how to code the project."

"More clear guidelines and expectations would have helped, and additional overview of the work that was needed to be done."

Q9, Cont'd

"With additional time, I would have known of various functional requirements before the development had begun."

Discussion

Also these are discussed by Section:

- ***Section A is about general feedback about the lack of requirements.***
- **Section B is about the definition of the SW development problem and the possible benefits from understanding the problem.**
- **Section C is about estimating the project's effort, cost, and time for development.**

Developers and RE

The evidence in this study indicates that developers and quality assurers believe in doing serious RE.

Most developers agree that requirements are very important.

All project members agree that RE improves understanding of details, dependencies, and complexities.

Project Managers and RE

The authors believe that, in general, project managers tend to skip the requirements phase because of pressure from senior management to be the first with a product on the market.

The Project's Manager and RE

Based on the FA's experiences at O and informal discussions with his fellow O employees, the FA concluded that …

The project's manager did *not* like requirements!

The FA drew this conclusion from the manager's behavior.

Manager's Behavior

Manager seemed to resist any suggestion of the development team's figuring out requirements collectively.

Instead, he asked each *individual* to build a prototype of a different feature, interacting with only the client, and reporting back to only him with completed prototype.

Phony Agility

Agility in sense of continuous interaction with client, …

but not in sense of communication with entire team.

Manager's Behavior, Cont'd

The FA believes that the manager associated knowledge with power and job stability.

If the manager is the only one that knows something, he is indispensable.

A requirements specification gives this knowledge to everyone in the project team.

Thus, a requirements specification is very low priority to the manager.

Manager's Behavior, Cont'd

Ironically, the lack of a systematic, coordinated attempt to determine all requirements up front →

the manager probably knew no more about requirements than did the team collectively.

Lack of RE and Productivity

In the absence of well defined requirements, productivity was hampered.

The resulting rewriting wastes time.

Apparent Outlier

One project member thought he would have spent less time in the requirements phase.

The reason is that he was worried about the architecture.

Past experience shows that some junior developers tend to worry more about the architecture or deliverables.

Discussion, Cont'd

Also these are discussed by Section:

- **Section A is about general feedback about the lack of requirements.**
- ***Section B is about the definition of the SW development problem and the possible benefits from understanding the problem.***
- **Section C is about estimating the project's effort, cost, and time for development.**

Importance of Understanding Requirements

When asked for the importance of understanding requirements in the different phases of the SDLC, the project members seem to believe that understanding requirements is relevant for most phases but documentation.

Importance, Cont'd

The reason is that developers were not forced to document their work. As a result, they do not value this phase.

Nevertheless, understanding what to do on design and coding seems extremely relevant since rework is lessened, the effectiveness of communication increases, and developers are able to make better decisions.

Importance, Cont'd

The fact that the amount of rework on this project was about the same as on other projects reveals that developing SW without understanding its requirements was the norm at O.

Regardless, the reality is that time, effort and cost of doing some tasks twice, or more, due to the lack of understanding requirements, is always a setback.

Productivity

The development team was actually quite productive when measured purely by lines of code produced per hour, i.e., gross productivity.

However, when measures by lines of *delivered* code produced per hour, i.e., net productivity, the story was totally different!

Lots of code was thrown out.

Productivity, Cont'd

Initially, the development team did not understand the product. As a result, the product quality was affected.

Over time, the development team was able to gather enough knowledge to develop the product.

Consequently, the overall product quality was acceptable. Nonetheless, the finished product was finished later than required.

Discussion, Cont'd

Also these are discussed by Section:

- **Section A is about general feedback about the lack of requirements.**
- **Section B is about the definition of the SW development problem and the possible benefits from understanding the problem.**
- ***Section C is about estimating the project's effort, cost, and time for development.***

Improving Estimation

The development team clearly indicated that RE can improve SW development and project management estimations.

When asked why discrepancies might exist between estimations and actual, project members opinions were very similar.

The common complaint was that the work had to be restarted due to missing specifications.

Improving Estimation, Cont'd

Discrepancies occur when the team does not work together from the beginning.

However, when working as a team the estimates are typically fairly accurate since everyone knows how long each piece should take.

When asked how developers would have improved their estimations the message is clear: the team needs a better understanding of the requirements and their scope.

Quality Assurance

The Quality Assurance (QA) Team started to test the product. They compared the old product against our newly created product.

If the new system was missing some functionality, or bugs were found, they opened a ticket in a bug-tracking application BTA.

Quality Assurance, Cont'd

By the end of June 2010, the QA team has logged 681 tickets.

Large number, even for O.

Origin of Tickets

For this CS, the FA tried to determine the origins of the 681 tickets.

After reviewing only the first 100 tickets, he gave up, confident of a representative sample.

37 of the 100 were from missing requirements, and …

the remaining 63 were bugs introduced during programming of known requirements.

Quality Assurance, Cont'd

The senior management team that approves the company wide deployment of new products had a hard time approving the project for continuation.

The main reason was that they thought the application was not robust enough based on all the bugs that were registered in BTA.

Transience of QA Team

After the development team spent nine months, October 2008–July 2009, sharing knowledge with one member from the QA team, he was moved to another project.

Consequently, we had to start from scratch with a new member from the QA team.

Ripple Effects

There were times when developers fixed one problem, and other functionalities of the application stopped working.

The Positives

In our case, we had the old system that we had to replicate.

Consequently, we could see how the old UI looks and duplicate it, with our new components.

Therefore, we did not waste any time trying to come up with a UI, saving some time.

DB2 Database

The old system's relational database is IBM's DB2. In the beginning stages, this was a problem, because no one in the team had the expertise on DB2.

However, knowing DB2 was not necessary, since the old system writes the queries into a log file.

DB2 Database, Cont'd

As a result, we had all the necessary queries to load into the UI components to get the database to perform the different actions such as: edit, save, and submit.

The Final Stages

Even though the developers were able to play with the old system, requirements were still missing. For example, if some fields were filled, other fields were automatically marked as required.

Consequently, a lot of tickets were opened about these kinds of issues.

Missing the Obvious

After working for two years on the same Webpages, developers started to miss the details.

Usually, other users, new developers, or clients saw things that people involved in the project for months could not see.

Doing It Right

After completion of the CS, near the end of the project, we tried to make things right.

We tried to establish genuine Agile SW development techniques.

Doing It Right, Cont'd

Every week, we went through a list of bugs and missing requirements, and tried to fix them.

Finally, on Fridays we deployed our changes. The QA team reviewed them and communicated the results to the project manager and the development team.

But No Credibility

Although, the project started to move at a faster pace, at this point it was too late.

The project manager lost credibility with senior management.

Project members also suffered, as no one got salary raises or promoted to higher positions.

Several developers quit, including the FA a bit later.

The Road to Improvement

How could we have improved the course of this project? Hofmann and Lehner state that there are three factors that contribute to project success: knowledge, resources, and process.

It is important to have the experience and expertise on a team to achieve effectiveness. However, there's always a dilemma with knowledge.

Knowledge Dilemma

Hofmann and Lehner identified the thin spread of application domain knowledge as one of the most salient problems in SW projects.

As explained, in this project, the manager associates knowledge with power and job stability.

He believes that if he is the one that knows everything, he will be needed at all times and be indispensable.

Resources for RE

Patricia Guinan gives an average RE team size of 5.6 members in the 66 projects she studied.

It is safe to say that the more people allocated to well-defined RE, the higher are the chances of your project to succeed.

Resources for RE, Cont'd

In our case, the whole development team was small.

Consequently, it is unrealistic to have five people fully devoted to RE.

However, if the project manager had had all project members thoroughly exercise the legacy product prior to starting development, the entire team would have understood what the new product had to do.

The Home Stretch

Our RE process had gotten to the point that project members could account for stakeholders' learning curves, and for any requirements negotiation.

Our architecture was able to support most of the requested changes, even if the requirements were always changing.

Enhancements Easier

Additionally, project members easily expanded the Website forms to comply with recent changes and new standards that the Canadian and United States governments had instituted for SW in PX's domain.

An Impact on Later Projects

Surprisingly (at least to these authors), the manager in charge of PX was given the responsibility to lead a new project for a new client.

This is another sign that the history of the project to build PX was well within the norm at O.

New Behavior

Nonetheless, before he quit, the FA was already noticing changes in the behavior of the project members and manager.

In a project member's own words: "moving forward, we need clearly defined requirements since they would have saved more time and efforts in terms of reducing the back and forth communication in the development and testing stage."

New Behavior

The new client was required to write a business requirements document which describes: business requirements, business policies and regulations, non-functional requirements, business data models, and much more.

New Behavior

Knowledge was not withheld anymore, as the business requirements document was frequently updated and distributed across developers.

The new client was expected to fully cooperate with the fields and search criteria that will be shown in the user interface.

Quelle Surprise!

At times, it was hard to believe that this was the same project manager and team that developed PX.

Evidently, the questionnaire made project members aware of RE.

Thus, the FA's case study seems to have induced a sustained after-the-fact Hawthorne effect.

Sustained Effect?

The FA has checked with his former teammates on the PX project and has determined:

As of July 2011 (about 12 months after conclusion of the survey), the improved RE process continues.

Evidence of Sustained Effect

One former teammate said:

“We are trying to deliver a product but the client kept changing it … but it [the process] is still improved.”

So far, there appear to be fewer defects, but the “product is still not finished so it’s hard to tell.”

Finishing by Deadline?

We are still ‘‘on track [to complete by the deadline] because of some padding we have been giving ourselves. We are doing a better job on this one.”

The customer seems to be satisfied. “They have been pretty happy with what they’ve been seeing so far.”

Even the Outlier Changed

The junior developer who had reported that he would have spent less time in the requirements phase …

approached the FA to ask if he could change his responses to the questionnaire!

(Of course not!)

But, even he had gotten the message of the importance of requirements.

A Possible Prescription

Perhaps, a good way to get an organization to improve its RE process, particularly when it has none is to …

Conduct a survey like that used for this CS in the middle of a challenged project, when it is apparent to *all* how challenged it is

even if no one admits out loud that it is challenged!

Similar Effects Elsewhere

As a result of comments by Daniel Schwabe, we discovered that others, in social sciences and medicine, had observed similar sustained, beneficial Hawthorne effects of randomized-controlled trials.

Some, e.g., McCambridge asks “Does research participation motivate behaviour change … and does it matter if it does?”

Hawthorne Effect by Design

Some, e.g., Feil, et al. even suggest "Intentional Use of the Hawthorne Effect to Improve Oral Hygiene Compliance in Orthodontic Patients".

Conclusions

This presentation reported the results of a CS that investigated how the lack of requirements affected one SW development process.

Data from the study show that the lack of requirements and requirements understanding had a direct impact on design, coding, testing, cost estimation, and project management.

Conclusions, Cont'd

That a product can be developed with little or no RE follows from the fact that we were able to develop the product without any serious RE.

But the process was very painful.

What remains constant is that RE can make the whole process better.

Conclusions, Cont'd

Getting industry to regularly do strong RE for its development is a cultural change that will take years.

Convincing management the value of RE is not easy.

Management is averse to trying unproved ideas that look costly and delaying.

Conclusions, Cont'd

Perhaps, just getting people to speak about unspoken and unspeakable difficulties they face and …

making people aware of other possibilities can help speed up the cultural change

Why the Hawthorne Effect?

Julio Leite wondered why the employees of O, all computer science graduates, had not applied what they undoubtedly learned in SE classes about the importance of RE and the effects of not doing it.

A Possible Explanation

Perhaps, all this stuff about proper RE and SE is just dry, useless, academic lecture material spouted by a prof …

who clearly does not know anything about the real world, …

who teaches SE because he or she cannot do SE, …

who preaches systematic methods that just don't work in real-life SW development trenches.

A Possible Explanation, Cont'd

The connection between what is taught in class and real life remains not apparent even at work, because no one mentions that what is happening is bad.

This survey drove home to each participant that what his prof said was *his* real life.

Once the connection was made real, and the symptoms were recognized as what the prof described, the remedies the prof mentioned became real.