ZAIA-Zero Adoption Impact Applications- A Research Proposal

By Karl Reed

Rev 2.5[1]

Synopsis

An overarching issue which the author believes will be central to the effective use of what will be the future generation of information technology reducing the cost of adoption of business processes and systems. The comments are not in themselves new, however, they emphasise applied research directions that will lead to more efficient software development processes. In many countries, the demography of industry sectors often means that there are extremely large numbers of small companies with between 1 and 10 staff. For these companies, adopting new processes, whether IT or non-IT, which either require very extensive training or disrupt existing work practices, is not acceptable. It may also become a barrier to even considering the purchase of IT products.

This research is intended to define strategies which will lead to the creation of products which have either Zero Adoption Impact or a very small impact.

The outcome may not be "formal", however, it may lead to new approaches to software development practice.

1. Introduction

In 2003, while working at the Frauhofer IESE, the author was prompted to propose the ZAIA concept, that is, Zero Adoption Impact Applications.

The basic issue here was that for millions of private users, negative adoption impacts are disruptive, and for the tens of thousands of very small companies that make up many industry sectors, the negative impact of new processes can be economically devastating.

In simple terms, a commercial organization of say three people will have trouble recovering the cost of a new "process" acquisition which, say, caused the losses of 6 person months of billable time in a single year.[2] As another example, we also have large organisations who adopt "whole of enterprise" applications which, on anecdotal evidence, cause large-scale disruptions. Concrete examples are the Australian Customs Integrated Cargo System (ICS) [Booze-Hamilton 2006] whose adoption caused massive problems on the Australian docks in October of 2005.

---

[1] Revised 11/5/2021 Rev number corrected, typos correctec
Revised 16/5/2016 "reference" to ubiquitous computing added.
Revised 11/3/2013 minor typos corrected, especially Table numbers
Revised 8/12/2012 corrected author list
Revised 27/2/2009 to correct layout of tables.

[2] I don't have exact figures for this, it is a supposition.

The objective of this research is to make a substantial contribution to the understanding of the problem, i.e, why we have this happen, and, to propose methodologies that will allow developers to produce Zero Adoption Impact Applications.

This work is clearly related to other work on Ubiquitous computing, however, approaches the problem from a different point of view, addressing both "visible" software systems (those users are of), and, the invisible, implicit, extension of functionality of existing software already being used

2. ZAIA Systems-Why We Need Them..

We have also seen an increase in the number of web-sites that provide end-users with services previously available from specialists. E.G. airline reservation systems, and end–user electronic banking to name just two. However, problems exist with overall quality of such systems. It is possible to spend several hours installing and coming to terms with such systems. (The scale of the problem has lead to additional ways of looking at systems design, so-called "user centred" approaches).

Taking a "high-level" view, one would expect that the characteristics of a customer base will have some impact upon the total investment that can be made in adopting a new technology. In principle, all businesses, no matter what their size, must obtain some kind of ROI for their IT investments. However, there are differences where very small businesses are concerned.

This can be summarized below:-

1. We want extremely low adoption costs
2. Where possible, new services should be re-arrangements of existing services already available
3. Must be extremely reliable, dependable and secure
4. Have extremely high rates of ROI for the purchaser
5. If possible, have very low cost of production/development

3. Evolving Processes- A related issue

In practice, we believe that processes involving human actors evolve as the human actors obtain experience and become expert, acquiring new knowledge about the process, its environment, its inputs and outputs and the opportunities for stream-lining. This is a ZAIA related issue, since the capacity of a process to fit, easily into an existing environment, may depend upon its ability to evolve to meet a users actual needs.

Table I General Comments

Issue	Requires	
Effectiveness of ICT investment	-Ability to justify ROI on ICT investment, and to protect existing investment -Protect and guarantee ROI	
Business Process Alignment extended to include adoption impact in general.		

4. Impact of TTM Product Development

Currently, we suggest that it is not easy for firms to assess the impact of new products produced in a TTM mode. The author has made the comment that "Time to Market is shorter than Time Design and shorter than the time to assess the business case for the new product." There are common cases where this will not be the case. If the product is developed as a result of an user's perception of a new product need, for example, it seems likely that the ROI issue will have already been resolved. However, if the product is generated by a supplier (of products) then it may be that users of existing products cannot keep up with the release frequency.

However, if the process development strategy was ZAIA conformant, then new products could be adopted without the problems above.

5. A ZAIA Driver.. Industry Demographics-Some Characteristics of the Financial Services Industry in Germany

The GfK report shows that there are large number of small organisations in the German financial sector. Table III is taken from page 57 of this report.

Table II Demographics of the German Financial Services Industry[3]

Company Size	No. Companies
1<9	91,711
10<49	2,474
50<199	1,401
<=200	1,943

[3] We assume that there are many industries in any country for which have this kind of demographics, although the precise industries will vary from country to country.

The very large number of small companies suggests "cost of adoption" as well as "ROI" will be important issues in the future take-up of IT services. No-one can afford to waste time adopting software, but, this is especially true for very small businesses. The author has made a preliminary assessment of the time to breakeven given exponential learning curves[4] and a particular productivity gain for any new process or tool. The analysis did not include acquisition cost, and the results were normalised using the "time to learn", which is accepted as the time taken learning to be reach 95% completeness. Table V below shows the result.

Table III Break-even Times vs Productivity Increase(Reed, 1999)

Normalised Time to Breakeven-No. learning times T/T_c	Productivity Increase Required
1	50%
1.5	30%
2.00	20%
2.5	15%
3.33	10%
7.00	5%

A rhetorical question would be "How often do you have a new technology that will give a 50% productivity gain?".

An equally important question would be "What properties should a product have to have either high productivity, or, a very short learning time, or both"

The author suggests that "Ease of adoption" is an appropriate design goal which addresses. By this we mean that an new product should have be able to replace an existing way of doing things, leaving as much of the human process as possible in place. That is, unless there is some (other) reason for changing that process.

By way of an example, one can consider simple credit card transactions at a supermarket checkout. These invariably require more "activity" on the part of both the customer and the sales person than the use of cash. However, the Hong Kong public transport Octopus smart card (which uses an induction loop, and whose on-card cash-balance is debited automatically when the card is placed against a sensor) requires LESS "activity" than cash[5]…...

Other researchers have already recognised there is a bias needed in software development towards user-centred and "value" based SE. I am suggestion a particular additional bias, which is that for small business and some other cases, the focus needs to be on ease of adoption.

[4] This is unpublished since it is elementary calculus, and simply part of lecture material in a software tools course.

[5] The card can be read through a wallet or purse… the customer doesn't even need to produce ut.

6. The Accounts renewal problem.

In a real world case, an organization introduced an new “accounts renewal’ process that tightened previously existing, but not enforced account-holder requirements. In the past, these were over-ridden, and completed later. The important thing was to get the account holder into the system. The new process required that ALL requirements be met, and had no provision for over-riding them. The company concerned had serious problems and risked loosing customers as a result.

This is a ZAIA issue, and one that should have been picked up by the analysts doing the pre-introduction analysis for the new system. However, it was not.

Its even possible that some stake-holders agreed to the change… they overlooked their actual work practice.

ZAIA approaches should be able to address this kind of problem.

7. Impact of Overarching Factors on Research Agendas…

The table below shows some over-arching software engineering issues and their relationship to ZAIA. They are not necessarily a direct part of the research, but, are relevant in that they may contribute to a total understanding of ZAIA development approaches. It’s the case that some of these have business process analogs.

Table IV-Research Issues and Sources for ZAIA Methodology Development

Overarching Factor	Implications	Research Area	Detailed Topics (notes)	ZAIA Issue
Preservation of legacy Systems	-Develop new services using existing functionality -Reengineer to improve reliability, performance etc. -Migration to better cost-performance platforms	-Identifying existing functionality within a legacy system and making it available via an API - Re-engineering - Performance engineering - Interpretive systems	-(User centred SE IS community may deal withy this already)	-Users may be able to work with previously known functionality
Re-configurable applications	-Ability to produce new applications re-using existing functionality -Retrofitting	-Identifying existing functionality within a legacy system and making it available via an API -Identifying basic functionality which is isomorphic across domains for reuse -Techniques for evolving systems -Reengineering	-evolvable systems -Dynamic Functional variation -Product Line strategies -Architecture recovery	-Ability of system to adapt to user learning
High Adoptability	-Business function oriented applications -Usability, HCI -Ease of learning -Dependability	-Domain specific processes -Usability/HCI -	-(may already be covered by IS researchers?) -user centred design processes -HCI centred design processes -Business process capture	-The fundamental ZAIA issue
Low cost	-Reuse -Application generation			
High usability	-HCI -Business function oriented	-User centred SE -Domain specific processes		
TTM	-reuse -productivity issues -process -design efficiency			

8. ZAIA projects-Zero Adoption Impact Applications

The table below shows some examples of ZAIA class applications. Some of these actually exist, or have been considered at some time.

Table V– Potential ZAIA Applications

Application	ZAIA Issues	ZAIA Challenge	Acquired Learning expectation	Acceptable training
-Wind-shear indicator for aircraft	-Current technology easy to read	-use new technology to produce easier to read display	-low	-low
-Automobile Controls	-Current very easy to learn and use -very expensive, electro mechanical	-Preserve existing metaphor -use cheaper technology	-low	-as current or lower
-Aircraft instruments	--Current very easy to learn and use -very expensive, electro mechanical	-Based on existing metaphor -use cheaper technology -better displays	-high	-high
-Legacy University Accounting system	-current technology known -work arounds assumed -evolved process development -used by hundreds of non-expert users	-improve application -limit change to user behaviour -identify combined process-actor activity and allow for this	-low	-low

Table V – Potential ZAIA Applications (cont'd)

Application	ZAIA Issues	ZAIA Challenge	Acquired Learning expectation	Acceptable training
-Tab-base	-current technology known -work arounds assumed -evolved process development -used by hundreds of thousands of non-expert users	-introduce totally new functionality -derive this from current use without commands -	-low	-low
-web-based airline reservation system for travel agents	-reduce cost of delivery -location-independent access	-existing system used by skilled and experienced operators -intelligent screen responses -v.low high through-put rates without stressing operators	-high	-medium
-web-based airline reservation system for end users	-reduce load on travel agents, allow direct selling -location-independent access	- system used by unskilled and inexperienced operators -do not waste clients time. Must be "better" than dealing with a travel agent	-v. low	-v. low

8.1 Tabbase- Integrated word-style table and data-base application

The proposal for a special tool for handling tables in word documents is based upon some experience and some assumptions about making the whole process "learning free". In this case, we assume some elementary knowledge of DB searching and structure, but want to avoid the need to actually specify the DB. I am not sure of how much of this is already in Word's table handling (my preliminary investigation suggests that it is not.. the use of DB's directly as tables is not itself new, however the objective here is real simplicity..).

This tool is based upon the manner in which I (and I assume lots of people) use tables in word. this idea is that each row in a table is essentially a record in the normal DB sense. However, there are some special requirements..

To illustrate, I've included a sample of a table.. I've put the text <cr> where the line-feeds are.. In this case, The table actually represents a network, which could be stored and interrogated if held in a DB. However, the user doesn't want to have to define the schema. Other examples exist, say where some student complaint is being tracked, where the entries in multiple columns form sub-records (I can show you an example).

Table VI-Example I, Developing Generalisations

Questions	***Issues***	***Sources***	***Possible Outcomes***
Q1. How can we develop and use generalisations?	*I1.1.Need appropriate types of specification and design statements*	*S1.1 Software specification and requirements fields*	*O1.1.1 Readily generalisable specifications* ***<cr>*** *O1.1.2. This is related to more general issues of spec. languages and****<cr>***
	I1.2 need strategies for generalising specifications	*S1.2 1.existing work in adaptable systems and generalisations, Reed;s work on isomorphisms, others on analogs****<cr>*** *S1.2.2 research into processes used by domain experts in design****<cr>*** *S1.2.3 research into psychology of idea associations****<cr>*** *S1.2.4 Work on evolvable systems****<cr>*** *S1.2.5 Study exactly how domain experts are doing this*	*O1.2.1 strategies for generalising specifications-processes****<cr>*** *O1.2.2 rules for identifying isomorphisms*
	I1.3 need to be able to recognise generalisations	*S1.3.1 research the literature and develop field further- IESE client companies system examples****<cr>*** *S1.3.2 examine the role of design reasoning recording as an aid.*	*O1.3.1 rules for describing specs that make identifying generalisations easy****<cr>*** *O1.3.2 design processes that generate generalisations, emphasising the design recording*

Using a table in this manner is pretty natural. You don't want to keep creating new rows, and, in any case, the sub-rows often have a common "root". Plus, there could be a number of "null" entries, where the user has aligned entries in different columns up. (This may be a bit tricky to sort out…).

Having the table properly stored in a DB would allow sensible searches to be made, and data to be extracted sensibly.

THIS PROJECT.. the goal is to design and demonstrate such a tool. Literature on the use of tables and their automation would need to be surveyed.

Things to add..

a/ types of actors, experienced, skilled, unskilled, replaceable.. impact on zaia

b/ (Barbara Russo raised the impact of adoption planning, and the use of champions and experienced trainers. My view is that this important, and we can develop a strategy for telling whether a proposed process can actually be economically adopted by this means. This needs metrics..)

c/ Research directions table