

Scrum Master's Scrapbook

or

(Wicked Agile – and my part in its development)

Wicked problems are never static and are subject to multiple interpretations. Peter Checkland

Author [Zak Moore](#)

Date 20-June-12

Version 1.0

Top tip: Use [Ctrl] Click to follow links. For internal links press [Alt] ← to go back to the main text.

Note: some diagrams and pictures appear fuzzy – that is to save space and protect the innocent.

Table of Contents

Scrum Master's Scrapbook	1
Introduction for busy people	4
Context.....	4
What did we do?.....	4
Establish, Motivate and Empower	6
Induction	7
Culture.....	7
Climate for innovation	8
Team roles.....	9
Personality styles	9
Learning styles	10
Reflection on emergent work practices.....	10
Learning Styles	10
Personality Styles	10
Team roles.....	11
Climate for Innovation	11
Culture.....	11
The best laid plans...aft gang agile agley.	11
Plans.....	11
Retrospectives.....	12
Reports.....	12
Knowledge Transfer is a misnomer.....	13
Domain knowledge	14
Traceable changes.....	15

Pain Points	16
Intra Team.....	16
Stakeholders	16
Onshore/Offshore.....	16
Lessons Learned.....	17
Foundation Theories.....	18
Causal loops provide more insight than scrum plans	18
Strategy.....	18
Managerial	18
Work.....	19
So What?.....	19
ISpace in one page	21
Ghoshal	22
Industrial age vs. knowledge age management	23
Project Pivots	24

Introduction for busy people

In this scrap book I will set the context for a global software development project and the state it was in when I was engaged to help clear a bottleneck in the design process.

I will then outline the approaches taken and how reality informed the implementation to:

- 1 [Elicit user priorities and valuation from a de-motivated workforce.](#)
- 2 [Establish, motivate and empower multifunctional, multicultural collocated design teams.](#)
- 3 How [the plans and progress against them was reported](#) to project stakeholders.
- 4 Explain why [knowledge transfer is a misnomer.](#)
- 5 As way of a conclusion I will run through the [pain points](#) & [lessons learned.](#)

Context

My client was a prestigious global publishing house headquartered in Amsterdam.

The organisation has strategic plans that are suited to the company's leading position in an oligarchic market: publishing about 2000 scientific journals with several best in class in science and health.



In 2008 they decided to replace the Editorial Evaluation System. The wicked element is the world is changing faster than the replacement system can be written – like so many software projects. Things did not go to according to plan. When I was engaged the project had been in development for 2 years and had no working software...at least there was a pretty flow chart on the glass wall where the team could hide from the rest of the world. The received wisdom was that the client had been unable to articulate the design – and that was the reason for poor code.

The project director decided to change the “way they worked round here” by creating multifunctional multicultural collocated teams and restricting the time they had to complete the system design. This was a job for you know who... Super Scrum Master!



What did we do?

We had to start somewhere so we got on our bikes and went to the top floor of a tower in the middle of Amsterdam. We invited 24 Euro-Users and got a bunch of input from them.



As an aside: one of the best (and macabre) excuses for late arrival was given by the Parisian attendee: *“Sorry I am late – there was a body on the line at Brussels”*

I also learned that Zak translates to Dutch as scrotum – just as I wrote my name tag. Oh Balls! I thought.

The aim of the workshop was to produce enough user stories for 2.5 sprints. Subsequent workshops would fill in the remaining sprints as we progressed. It was not an explicit objective that a comprehensive list of user stories was required to act as a contractual baseline – at the start of the process. This created tensions later in the project.

We had a lovely view of Amsterdam so we covered it up by plastering post-its and story cards over the windows. T-shirt sizing (XS, S, M, L, XL, XXL) was used to help provide *relative* business value for each story. Ranking was carried out to achieve a semblance of priority by epic and stories within them.



During the day we used different group sizes (24, 12, 8, 4, and 2) to carry out tasks quickly and effectively. The meeting ran to time – to show that it was possible for things to happen. Most of the staff had been de-motivated during the interminable and fruitless build.



With the stories ranked by business importance it was time to fill in NFRs and acceptance criteria. I ran two sessions: the first created a burn up chart. The second session started by asking the teams to commit to delivering an amount of value in a fixed time. I then ran the second session as a burn down. This helped educate the users – our clients and customers - to agile processes.

We used a variety of techniques to elicit stories and acceptance criteria. As an example: Haikus were written to stimulate the reporting muse...

Without reporting	Is it really true?	I want to see stats	Show me the data
No clue where you're going	My reviewers seem so slow	how is my journal doing	I want numbers and graphics
Lost in the forest	Tell me what to do!	what can be improved	keep it simple please

The result was a roadmap consisting of a prioritised set of epics that each had user specified relative business values aggregated from elaborated stories complemented by non functional requirements and acceptance criteria.



With the business value taken care of attention turned to the technical work of design: the next step was to prepare for onboarding the design teams.

Two teams of eight had been specified. Each team would have: Subject Matter Expert [SME], Business Analyst, Process Technical Lead, Content Technical Lead, User Centered Design [UCD] Expert, Prototype builder and Test Analyst. The teams were facilitated by a product owner and scrum master (me). The teams were a mix of Client employees, Offshore software employees and contract staff. The nationalities involved were: Dutch, British, Irish, French, Japanese, American, Portuguese and Indian. None of them had worked in an agile way before.

I prepared a five layer induction course to cover the softer side of wicked agile as well as the mechanics.

Establish, Motivate and Empower

The teams were called Calvin and Hobbes. Together they were formidable – in a cuddly kind of way...



They came from everywhere to take the challenge. Yes Siree!



So we helped them get to know each other by culture, team role, personality & learning style. This approach is based on [Knowledge Economy](#) management principles. They complement Agile roles very well but are not limited to software development.

The driving force of transferring the user know-how by Subject Matter Experts [SME], acting as proxies, to information contained in the design attributes was based on [double loop learning](#). It was the intention that the tacit knowledge not written into the designs would be communicated to the offshore coders by the technical leads who were returning to India after the design phase(s).

Induction

This section may be larger than one would normally expect. I make no apologies: setting the foundations of empowerment is fundamentally important.

The aim of the induction course was to allow the teams to gel and provide them with the ability to design work practices that facilitated individual [flow](#) as described by Csikszentmihalyi.

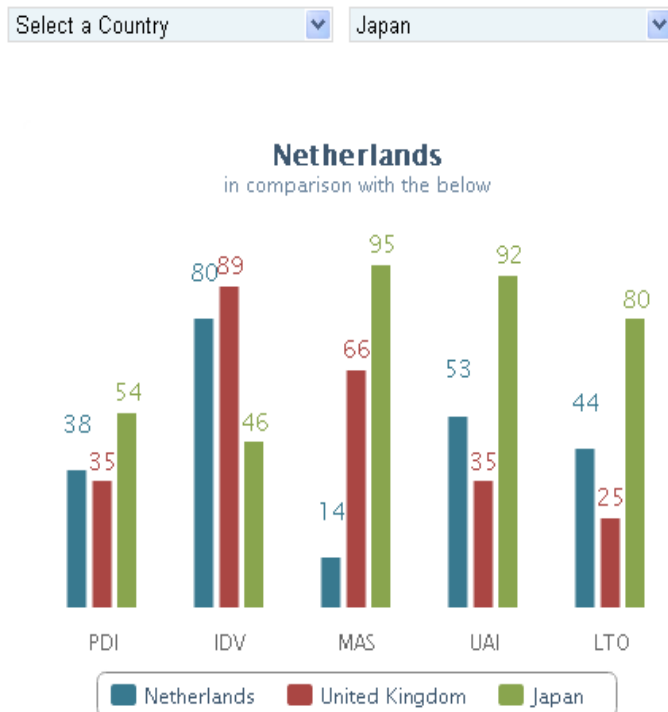
An outside-in approach was used to gel the individuals into teams. I started from cultural differences and drilled down to individual learning styles through climate, team roles and personality types.

Understanding learning styles means the work can be tailored so that individuals play to their individual strengths and collective potency.

Applying active listening and using a set of icebreakers throughout the day helped instil a collegiate atmosphere. We took the “deaf guy in the room” approach so that everyone could speak and be heard. The different accents took time for foreign ears to adjust to...This was an important precedent to set since there were many discussions to take place about understanding the requirements and then translate them to specifications that the offshore teams could use to produce the working codebase.

Let’s examine the elements of culture, climate, team role, personality type and learning style in turn.

Culture



With many different cultures to accommodate on the team I selected Hofstede’s cultural dimensions to encourage the members to explain their different ways of working. What we wanted was to encourage the Indians to question the information that was being presented so that misunderstandings could be ironed out and translated effectively for the offshore developers.

Hofstede has revisited his original work to incorporate a Confucius dimension to reflect China’s emergence in world commerce.

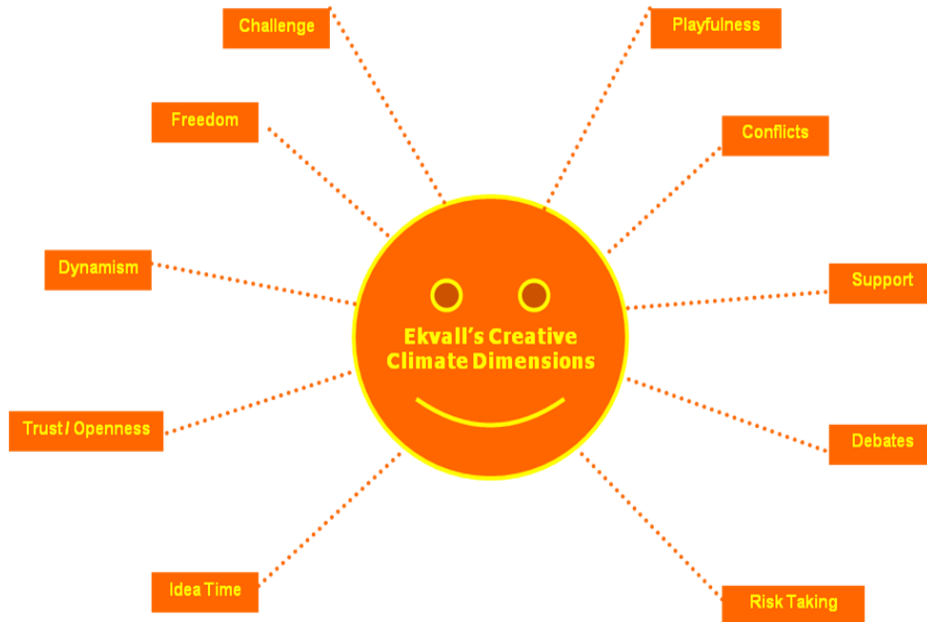
One curiosity of carrying out this global cultural analysis is that Geert Hofstede lived one stop down the railway line from the office.

The driving force of using Hofstede was to *leverage the differences in thinking* between team members to provide foundational learning along the lines of [Ghoshal’s Organising Framework](#) to complement the perceived efficiencies in developer arbitrage that had been unrecognised to date.

Climate for innovation

As a first action to empower the teams I selected Ekvall's creative dimensions. Each team voted for their own prioritisation of the governing variables. This was a first for the participants and showed them that the command and control structures were loose enough to allow environmental changes.

I believe that teams that are trusted to change their governing variable prove to be committed rather than compliant. The higher morale leads to higher drive ([See Causal Loops](#))



Each team dot voted for their preferred dimensions. Trust and openness scored highly for both. Hobbes was more for debates than Calvin. The red dots on the second picture are the inter-team dimensions.



Dimension	Hobbes (Green Dots)	Calvin (Red Dots)
Idea Time	0	0
Trust/Openness	4	4
Dynamism	2	0
Freedom	0	0
Challenge	4	3
Playfulness	4	0
Conflicts	0	0
Support	0	0
Debates	0	4
Risk Taking	0	0

With the environmental variables for innovation set and people understanding that there were different expectations we moved to look at the individual styles: team roles, personality and learning.










Team roles

Each member carried out an online Belbin test to show them where their strengths and weaknesses were.

One team had three completer finishers – and that was the team that had varied productivity and more stresses during the project.

Everyone enjoyed this exercise as it gave them the chance to speak about their favourite subject: themselves.

How do we explore that in more detail? The big 5 or MBTI could be used but...

	Team role	Strengths	Allowable weaknesses
Action oriented roles	 Shaper	<ul style="list-style-type: none"> Challenging, dynamic, thrives on pressure The drive and courage to overcome obstacles 	<ul style="list-style-type: none"> Prone to provocation Offends people's feelings
	 Implementer (company worker)	<ul style="list-style-type: none"> Disciplined, reliable, conscientious and efficient Turns ideas into practical actions 	<ul style="list-style-type: none"> Somewhat inflexible Slow to respond to new possibilities
	 Completer finisher	<ul style="list-style-type: none"> Painstaking, conscientious, anxious Searches out errors and omissions Delivers on time 	<ul style="list-style-type: none"> Inclined to worry unduly Reluctant to delegate
People oriented roles	 Co-ordinator (Chairman)	<ul style="list-style-type: none"> Mature, confident, a good chairperson Clarifies goals, promotes decision-making, delegates well 	<ul style="list-style-type: none"> Can often be seen as manipulative Offloads personal work
	 Teamworker	<ul style="list-style-type: none"> Co-operative, mild, perceptive and diplomatic Listens, builds, averts friction 	<ul style="list-style-type: none"> Indecisive in crunch situations
	 Resource investigator	<ul style="list-style-type: none"> Extrovert, enthusiastic, communicative Explores opportunities Develops contacts 	<ul style="list-style-type: none"> Over-optimistic Loses interest once initial enthusiasm has passed
Cerebral roles	 Plant	<ul style="list-style-type: none"> Creative, imaginative, unorthodox Solves difficult problems 	<ul style="list-style-type: none"> Ignores incidents Too pre-occupied to communicate effectively
	 Monitor evaluator	<ul style="list-style-type: none"> Sober, strategic and discerning Sees all options Judges accurately 	<ul style="list-style-type: none"> Lacks drive and ability to inspire others
	 Specialist	<ul style="list-style-type: none"> Single-minded, self-starting, dedicated Provides knowledge and skills in rare supply 	<ul style="list-style-type: none"> Contributes only on a narrow front Dwells on technicalities

Personality styles

	Known to Self	Not Known to Self
Known to Others	Arena independent intelligent knowledgeable logical observant	Blind Spot adaptable bold calm clever complex confident dependable energetic friendly helpful organised proud reflective relaxed responsive searching trustworthy warm witty
Not Known to Others	Façade ingenious	Unknown able accepting brave caring cheerful dignified extroverted giving happy idealistic introverted kind loving mature modest nervous patient powerful quiet religious self-assertive self-conscious sensible sentimental shy silly spontaneous sympathetic tense wise

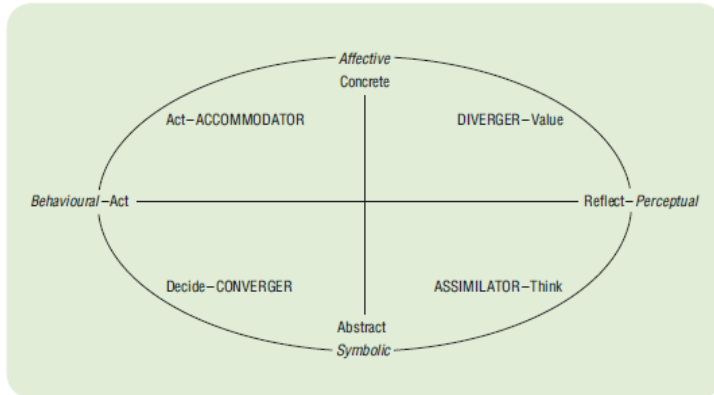
Personality tests can be considered as inexact as they are primarily self reporting. They are also reductionist in their approach and miss the holistic view of a person. They can be useful...but I prefer to use a JoHari approach (left) as it provides other people's views of the individual concerned. It can cause a sharp intake of breath to get started and may not be ideal in politically charged environments.

As Burns put it so succinctly: *"Oh would some power the giftie gie us to see the way others see us"*

This approach lends itself to a better understanding and can be taken by the individual as a way to achieve personal development. From the above chart I know that I have to work on tempering my complexity. It is an ongoing challenge – but enjoyable.

Reflective practice is paramount to making this technique work.

Learning styles



To loosely cross check my impressions from the previous exercises we took a quick look at the individuals' learning styles.

This allowed me to subsequently provide each member with activities that played to their strengths when they were blocked. As a scrum master of experience I know that **not every block I need to remove exists in the external environment.**

And finally I ended each induction course with a video of Dutton Engineering that showed agile, self-organising practice in an engineering context – and also explained that a good agile manager should be able to leave teams to run themselves within 6 months. I have achieved that.

Now let's examine how the theories translated into the day to day work – and the emergent practices.

Reflection on emergent work practices

Learning Styles

Instead of the teams working around a Kolb model of division of labour by learning-style there was a comprehensive involvement of all designers in each decision.

It transpired that the model used in practice was Lewin's Visual, Audio, and Kinaesthetic (VAK). The layout of the happy paths of wire-flows on the floor and the ability to walk through the system was useful in creating shared mental models.



People liked this approach. Getting out of their seats was enjoyable.

Personality Styles

One team was well balanced with the normal fault lines between Feature Driven Design [FDD] and UCD. This is to be expected. Tension in design is fine – so long as it resolves like a good tune.

In contrast the stresses experienced by the workers who were married with kids and had to leave their families may have led to bolshie behaviour: it was sometimes interpreted as "overbearing" by other team members. A simple reminder about the initial vote for Openness and Trust and ensuring all meetings were held in an open office nipped the problem in the bud.

Team roles

The team that had the imbalance of complete finishers was more erratic in its approach than the more varied team. This is normal in male dominated design. Informally it is called tech pissing...That said they reported 78% against a benchmark for agility while the “smooth” team reported 68%

Climate for Innovation

It soon became apparent that the UCD could be developed faster than the FDD attributes (Use Case, Technical Design). Previous work could be reused by the UCD expert involved in the project.

The documents had to be kept in sync to keep the quality manager happy.

To keep the documents in sync the team devised and tested various ways of working and eventually settled on the ideal designer day – showing its variance over the sprint duration.



Culture

In hindsight I would have used a [21st century management versus 20th century management](#) comparison to set a collegiate tone - with Hofstede playing an auxiliary role. The offshore workers were used to a tight command and control working culture. The Dutch employees were so bound by the [polder model](#) everyone had to have their say before any decision could be made.

The best laid plans...aft gang agile agley.

Plans

Being gnarly I can assess the state of projects by intuition and experience. As a way of explanation my gut feel comprises 3 interlinked [causal loops](#) when looking at software development issues. This allows me to formally model and explain intuitive reasoning.

The first couple of sprints were used to establish a benchmark of achievable **technical** value. We had project planning poker cards made and used them. There were difficulties that we had to overcome:

Firstly the members from command and control cultures had no idea of being responsible for estimation or planning. It took a couple of rounds over the first 2 sprints for them to appreciate that they were not being held to account for the group votes. There was a lot of “follow the leader”.

Secondly the two teams had widely different measurement scales: one used small units (1 -5) and the other was much higher. A benefit of agile is that the *relative* nature of the planning allowed both to be accommodated in the separate work streams.

Initially I selected Jira to plan and control the work. It turned out lacking in several key areas:

It was too restrictive in its hierarchical breakdown.

We were implementing a new “way of doing things round here” and the prescriptive nature of Jira’s workflow was an anathema.

It did not show interdependencies between Epics.

Design is more explorative than coding and needs to aggregate many divergent items into an abstract and codified whole – rather than 1-2-1 user story to function mapping of programming.

Retrospectives

Product

Each sprint lasted 3 weeks. On the second week the teams presented a show and tell (product prototype review) to the clients and customers. Feedback was noted and incorporated into the design documents during the final sprint week.

Initially we had 17 questions. By the 4th sprint we received 2 questions. This shows that the teams managed to empathise with the users and provide prototypes that were in keeping with their expectations. I had a SERVQual gap analysis to hand if required – in this case it was not needed.

It was also beneficial to let the users and designer meet and mix over coffee and chocolate biscuits. It reminded each other that the people on the other side of the wall were human as well.

Process

On the last day of the sprints the teams carried out a process retrospective. The retrospectives were designed not only to do the normal agile review processes in terms of work done, rejected and deferred but also examine the governing variables that had been dot voted at the beginning of the project.



Initially I thought that the teams would want to change their climate variables at the end of the sprints. This was an incorrect assumption. They appeared happy to change the things under their direct control. I am open to relooking into this with Kanban style teams that have been working together for 6 months or more.

One exception was that we revisited was the openness and trust variable. At one point under the auspices of a new SME the teams took to meeting in private rather than in the team room. After a brief discussion they agreed with me that holding all meetings in the team room was the best way to ensure openness and trust. I adore the Brain Clough approach to management!

Reports

Stakeholder level

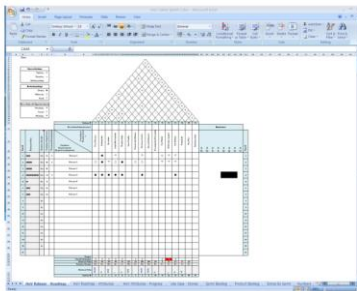
Playing what I saw I changed the planning and reporting mechanism from Jira to a variant of the [QFD](#). I implemented the house of value to report at 3 levels: project, sprint and intra sprint levels.

For the first time the interdependencies amongst the epics could be seen – as well as the strength of relationships between the phases in the project. A counter intuitive insight was that the newcomers

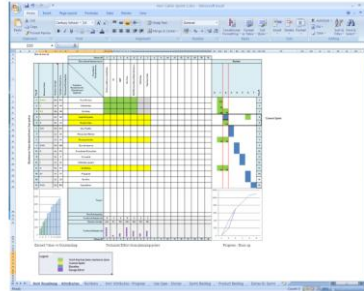
found it more difficult and time consuming to incorporate design that had been previously carried out rather than vanilla design.

House of Value:

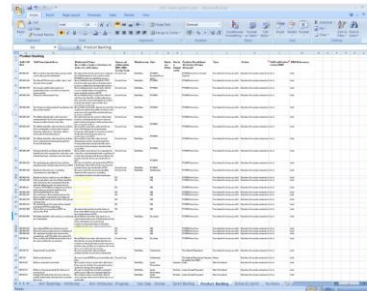
Roof – dependencies



Sprint Overview



Backlogs – showing scope changes



[A concise introduction is available on request.](#)

This technique may not scale to all solutions. A web version rather than the SharePoint version I hacked together would be preferable as it could have incorporated email alerts and daily updates at the appropriate levels.

This approach covers one of the big questions CTOs have to answer – How is technical effort aligned to business strategy? The sprint overview holds the answer by relating business and technical values.

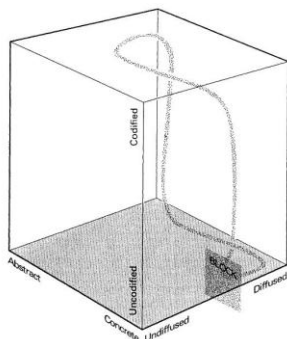
There were other problems in dealing with stakeholders –It is my personal view that they got caught up in trench warfare along corporate lines. They lost the focus of producing software and the corporate *raison d'être*: delivering shareholder ROI. There have been many personnel changes but the same problems persisted suggesting that the problems are systemic.

Knowledge Transfer is a misnomer

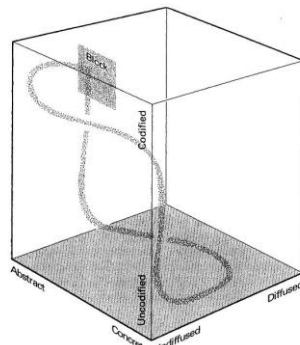
Knowledge Transfer should be called validated learning. Short of a brain transplant tacit knowledge cannot be transferred. I am unaware of any successful operations of this nature...

The key blocks in knowledge transfer validated learning can be represented by the [ISpace](#) and social learning cycle. The blocks that emerged during the project are shown below:

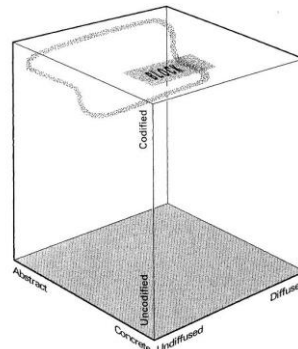
Design – could not create a baseline design



Knowledge did not transfer offshore



Coders were not trained to deliver



Initially I had been tasked to assist with removing the block on the scanning, abstraction and codification (design). It soon became apparent from informally eyeballing the low level design that there were other knowledge blocks: namely diffusion and the level of impact – lack of training of the offshore resources. This was further vindicated by a technical review carried out by Oracle.

With the team taking great strides to remove the scanning, abstraction and codification block I turned my attention to the other 2 blocks.

Domain knowledge

To ensure that the offshore team had an appreciation of the domain and subsequently the specific design they were working on the project communication officer devised a simple quiz.

So long as the quiz was not used as a selection tool – but as a learning aid there was support from the original project director. Subsequent staff changes sidelined the approach.

When the design team members from India returned to the offshore workplace they were assigned to other phases of the project - as I suspected they would.

Having had experience of the managerial stalling actions from the offshore company I had taken the precaution of having the team members create quizzes regarding the design attributes – use cases, design briefs, and prototypes for each Epic.

The screenshot shows a Moodle quiz report for 'Evisive Community'. The page title is 'Evisive Community' and the URL is 'comscentia.com/moodle/mod/quiz/report.php?quiz=1'. The report shows 12 attempts. The table below is a summary of the data shown in the report.

First name / Surname	Started on	Completed	Time taken	Grade/20	#1	#2	#3	#4	#5	#6	#7	#8
Hartbeje Branz	27 March 2012, 09:32 AM	27 March 2012, 09:36 AM	4 mins 9 secs	16	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25
Sam Weightman	30 March 2012, 11:57 AM	30 March 2012, 12:02 PM	35 mins 26 secs	6.58	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	0.93/1.25	1.25/1.25	1.25/1.25	1.25/1.25
Sheo Peak	11 April 2012, 09:48 AM	11 April 2012, 09:52 AM	4 mins 34 secs	6.37	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	0.93/1.25	1.25/1.25	1.25/1.25	0.93/1.25
roy Cooper	4 April 2012, 10:41 AM	-	open	-	-/1.25	-/1.25	-/1.25	-/1.25	-/1.25	-/1.25	-/1.25	-/1.25
Edward O'Brien	9 May 2012, 09:43 AM	-	open	-	-/1.25	-/1.25	-/1.25	-/1.25	-/1.25	-/1.25	-/1.25	-/1.25
Zak Moore	20 March 2012, 12:23 PM	20 March 2012, 02:34 PM	1 hour 1 min	7.54	0.93/1.25	0/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	0.42/1.25
Yarah Kirk	18 April 2012, 01:06 PM	-	open	-	0.93/1.25	0/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	0/1.25
Halewood King	11 April 2012, 03:18 PM	11 April 2012, 03:42 PM	3 mins 22 secs	16	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25
Hartbeje Toekler	15 May 2012, 02:54 PM	-	open	-	-/1.25	-/1.25	-/1.25	-/1.25	-/1.25	-/1.25	-/1.25	-/1.25
Isontine bark	21 March 2012, 09:57 AM	21 March 2012, 09:13 AM	8 mins 14 secs	6.21	0.93/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	0.93/1.25
Katli Bennett	29 May 2012, 01:22 PM	29 May 2012, 01:25 PM	3 mins	6.63	0.93/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25	1.25/1.25
Joost Willemson	26 March 2012, 03:40 PM	-	open	-	0.93/1.25	0.93/1.25	1.25/1.25	1.25/1.25	0.93/1.25	1.25/1.25	1.25/1.25	1.25/1.25
Overall average				6.82	1.00/1.25	0.90/1.25	1.25/1.25	1.25/1.25	1.11/1.25	1.25/1.25	1.25/1.25	0.93/1.25

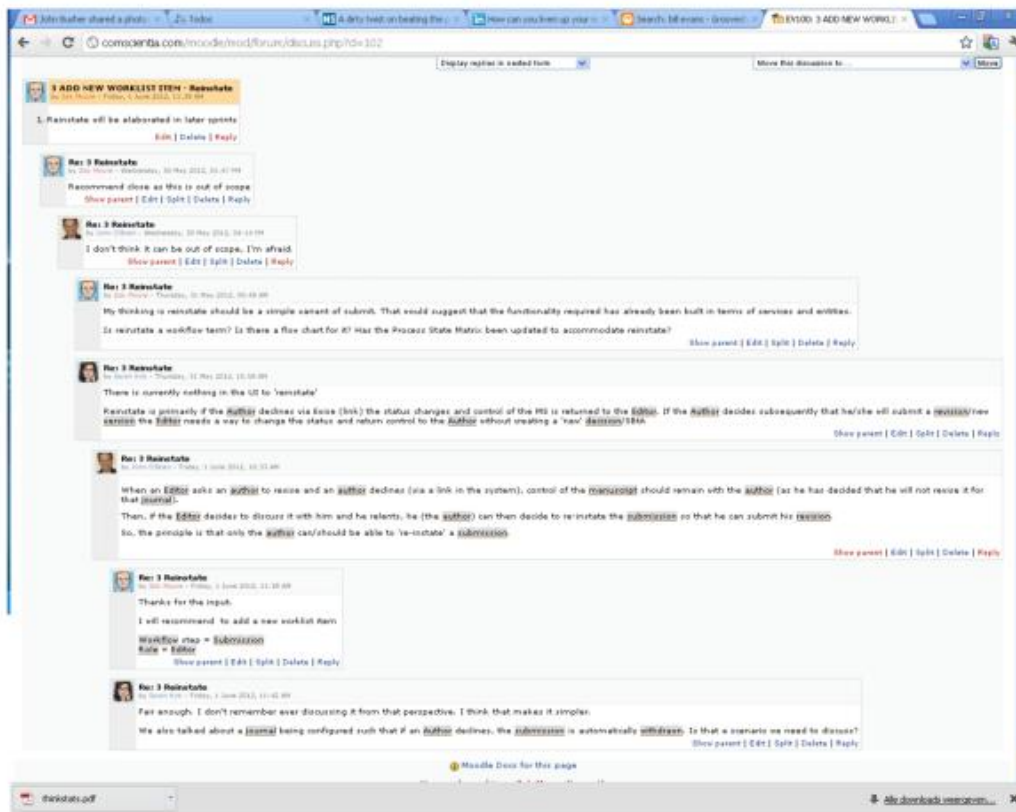
Below the table, there are options to 'Select all / Deselect all', 'Download in ODS format', 'Download in Excel format', and 'Download in text format'. There are also preferences for 'Show / download' options and 'Your preferences for this report' including 'Page size' (30) and 'Show / download marks for each question' (Yes).

These were deployed to be taken by other offshore workers to prove that they had read and **understood** the designs they would code against prior to writing their first unit tests.

The initial results were encouraging as the people who took the quizzes offered to write more questions for those that were to come later. When I tried to move it from guerilla mode to a formal method the stalling between both client and offshore development organisation was frustrating. It was about this time my teeth started to itch.

The quizzes were hosted on Moodle along with the project glossary. Later on I set up a set of forums to address the issue of open items in the use cases. This provided traceable discussions to inform the scribes of changes that needed to be made to their documents when they were retasked to this part of the project.

Traceable changes



To ensure that changes to the design output could be captured in a way that reflected the collocated process the remaining onshore workers discussed the open items on the use cases via a forum.

The feedback was positive and many people found that tacit information that had not surfaced during face to face discussions when everyone was together started coming during the online discussions. This was counter-intuitive – but welcome.

Wikis have been used by developers for some time. This more formal approach makes the discussions accessible to the non technical project stakeholders.

Pain Points

Every project has its pain points. This one had a few headaches.

Intra Team

In the team the expected pain points emerged: Most development projects have tensions between the front end expectations and backend capabilities. It is like the difference between mayonnaise and jet engines ([Herd, Mark Earls](#)). User Centered Design [UCD] is complex, like mayo, – it is a one way process. Feature Driven Development [FDD] work is like engineering: there are lots of interconnected components but they can be changed or used to reverse engineer the processes.

The self organization within the team resulted in them designing their own work days. This was a revelation to the offshore team members. Changing the governing variables increased the drive in the team – in comparison to what had been before.

Homesickness among team members adversely affected morale.

Stakeholders

Customers – who needs them? We did! We carried out UCD with show and tells that were well received. That said the technical delivery manager was annoyed that the “users now had expectations of what should be built”. You don’t need Mystic Meg to tell you where this is going to end up.

Over time the client had Fred Headed (lost the knowledge of) it’s IT team to such a degree they had nobody could read neither the *application* design nor code to determine the state of it. This led to tensions – and raises significant risk of the system not operating as expected. *Caveat Emptor*.

The clash between work methods in the client was troublesome. The project manager wanted a full set of user stories (700) for 30 epics so that an estimate could be used to set the fixed price on the delivery. The product manager followed the agile methodology and created backlogs for viable deliverables. The tensions affected morale. A more practical approach would be to do the design on a time based approach then fix the development costs. It was impossible to have sensible ideas stolen.

Onshore/Offshore

The attrition and variable quality of staff in any large software house results in client risk. We tried to reduce it through validated learning. OffShoreCo had to hire freelancers to cover the shortfall in staff. Thankfully the contractors that came in were of a higher calibre than the offshore workers and training was given.

Visa restrictions meant we lost team members after 3 months. Just as they got through the first agile plateau they were sent home. This stopped any chance of taking teams into Kanban processes and thus improving value chains between the on and offshore workers by establishing [flow production principles](#).

Lessons Learned

Strategy should be understood by all people involved in any project. The lack of an overarching shared mental model was telling.

Understanding **salience** is essential. Different stakeholder perspectives clouded the project vision because they were focussing on their small problems. The **systemic failure** to deliver had not been recognised so there was a continual changing of the guard – but the mistakes were repeated.

Allowing teams to **change their environment** needs practical alterations they can make. This is evident by the ideal designer day rather than ongoing voting to change the Ekvall dimensions.

Knowledge cannot be transferred. The closest comparison is validated learning. That said know-how can dissipate from the client and be gained by the offshore company. Will it be used competitively?

Each team had their own way of working. The workflow team was led by UCD and the toolset team was led by FDD. Different team configurations should be used depending on the nature of the work.

Validated learning is essential to ensure that the [deep smarts know-how](#) from the client has been successfully converted to information that is understood by developers prior to development.

The quizzes were enjoyed by the developers when run in guerrilla mode. There was resistance from the offshore managers. This is an example of **mistrust** between client and supplier. My read is that the offshore company had not trained its staff sufficiently. This **negates development arbitrage**.

Kinaesthetic learning was good. Everyone involved said it was informative to “walk the system”.

There is a danger that the middle managers will get more caught up in the contractual game – rather than focussing on delivering value to the client and their investors. (The role of [middle managers](#) has changed in knowledge era companies).

There are alternative methods such as *Design Thinking, Time boxed Ideation or the Buffalo method* that should have been considered for this project. Shoehorning a fashionable software development method (Scrum) that is appropriate in collocated teams into an offshore fixed-price waterfall project is not easy.

Homesickness leads to low morale. Test potential workers to determine if they can operate effectively when away from family for extended periods. This needs to consider personality and circumstances.

Assess the technical ability ([brainbench](#)) all participating staff prior to allowing them into the project. **Improve selection** – it is impertive in the knowledge economy.

From a standing start the benchmark for agile practice after 3 months was an average 73%.

Sanity Warning

If you find management modelling boring stop here: there is no point in reading any further. Go do something enjoyable instead.

Foundation Theories

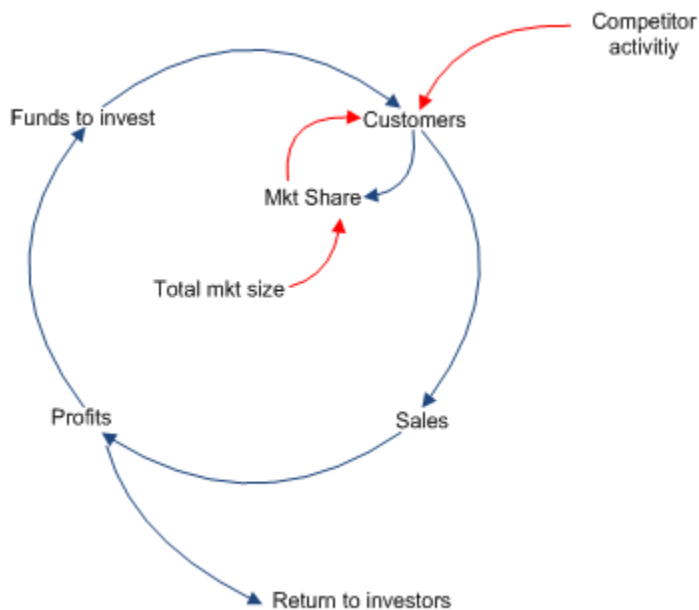
The theories that underpin the activities described above are taken from sound strategic thinking, common place models and applied to what I found as the project progressed. This section is more abstract than the previous ones.

Causal loops provide more insight than scrum plans

In keeping with the people before process approach and realising that the problems were being compounded I called upon a [Senge](#) like approach and used a 3 layer causal loop model.

Strategy

The topmost loop is a simple representation of any commercial business organisation when viewed from a strategic level. It exists to make money for the owners.



Investors expect a decent return on their investment. They can go elsewhere for it whenever they decide.

For example if an investor spent \$45M on a software project and it did not deliver working ware what action should they take?

One way of determining whether to pull the plug or carry on would be to examine

the lower level models and run scenarios. Let's do just that.

Managerial

Looking at the next level down – a very simple way of describing a project is it has a target, a plan of work and work that is actually carried out.

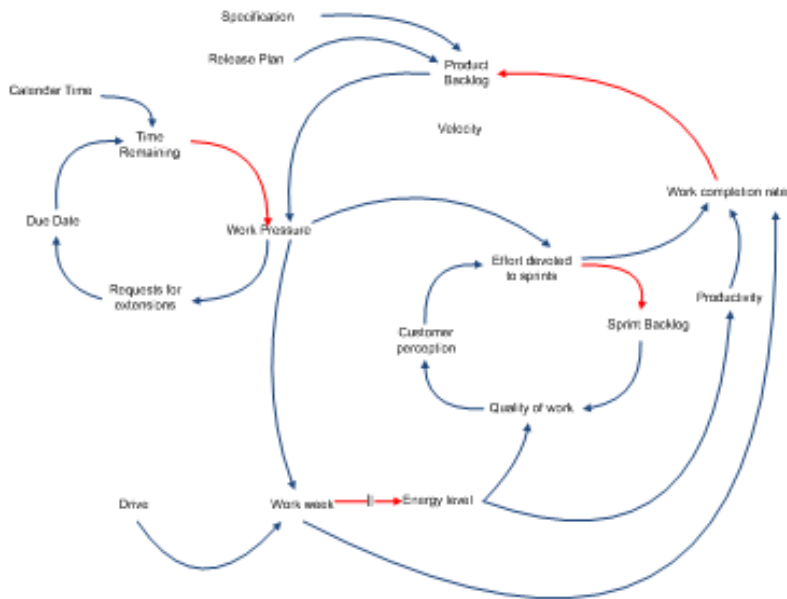
Experienced managers focus on the variance and not the plan. They know that not all functionality needs to be delivered to go live.

Bad managers continually stop and re-plan. What affects the actual achieved? Let's look...



Work

In this instance the model of designing was more akin to learning so I opted for **the dog ate my assignment** model. I like the analogy to a schooldays fear of late submission. It is also useful for fixed date delivery as it is akin to exams – the delivery date does not move. The pragmatic manager will know that corners are cut during design. So long as the corner cutting allows him/her to manage expectations of variance (above) the project can be successful.



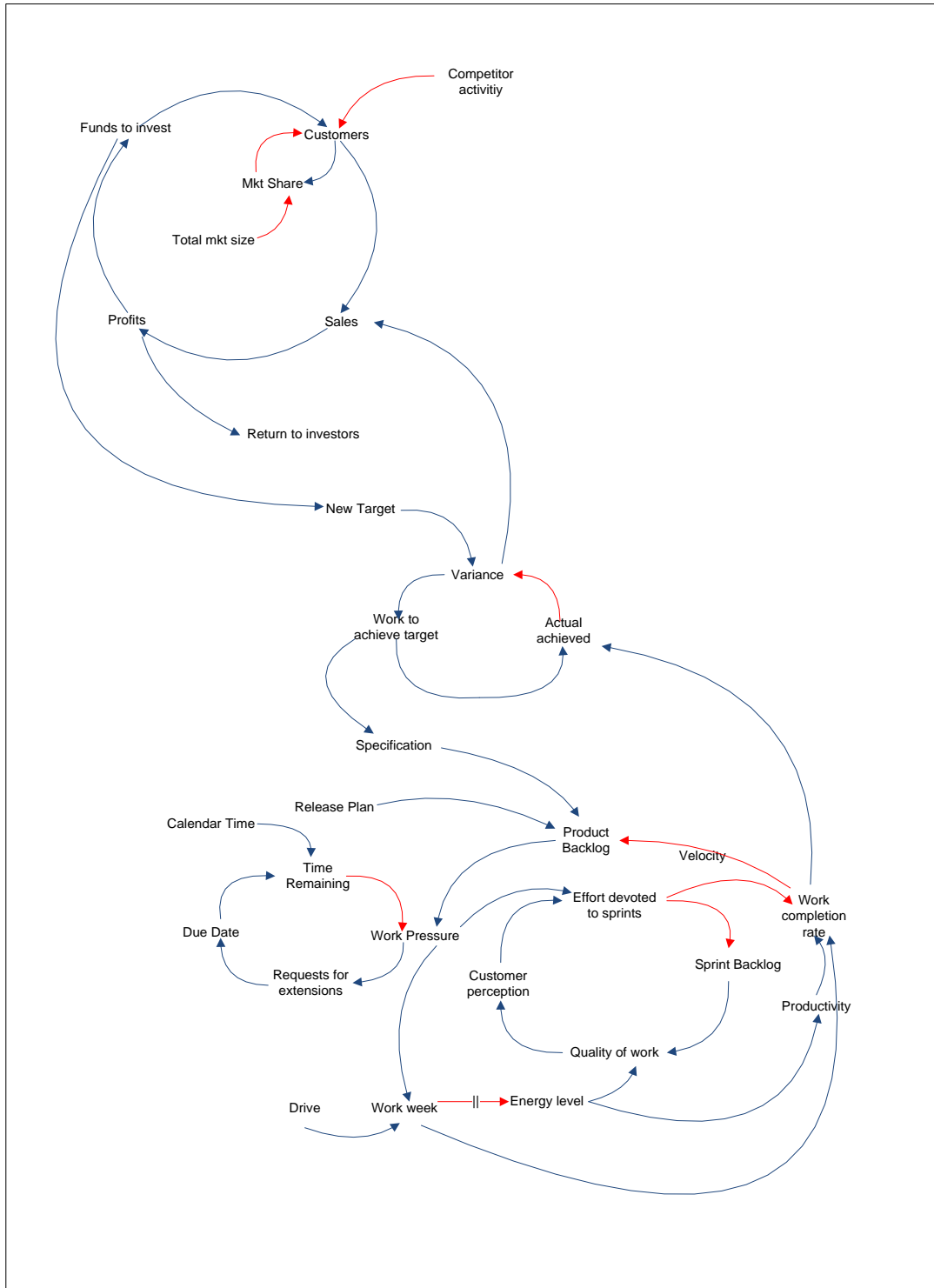
So What?

These loops are common in systems thinking. They may be new to *agilistas* – as they are a bit beyond most Scrum based theory. Complexity in Agile development is starting to gain traction. Jurgen Appelo is a decent starting point.

The models can be used through tools like Vensim to provide scenario modelling that is beyond the sophistication of most agile planning tools that I am aware of.

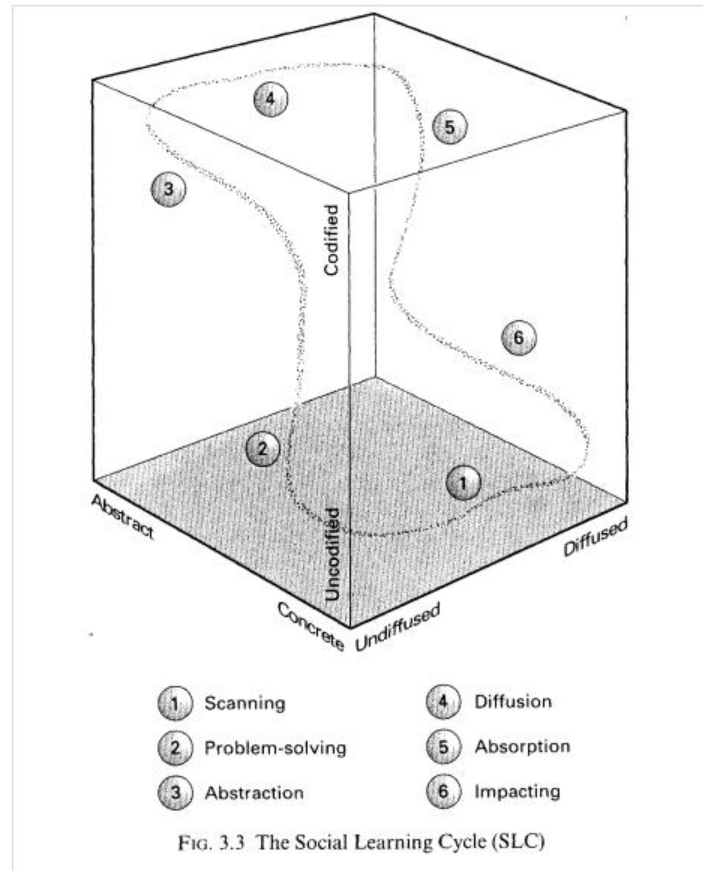
By having scenarios of how the small changes in projects compound into late delivery, managerial variance, market share and ROI it is possible to measure the project during execution and take timely corrective actions in the small.

As with all models and simulations they are not actual reality – they are very simplified views of the real world. They are however useful in preparing for the unforeseen. **Plans are worthless, but planning is everything – (Eisenhower, 1957)**



By joining up the loops it is easy to visualise how an action in the small can be compounded to affect the overall plans of the company.

Ispace in one page



The Ispace and SLC is a very useful model and a good reflection of the learning that was used in during the project. During the release retrospective the team declared it preferable to Nonaka *et al* and Cook and Brown. See [Knowledge Assets](#)

Initially developed by [Max Boisot](#) it has been developed by [Dave Snowden](#) into the [Cynefin](#) – which is the basis for the [project pivots](#) that are prevalent in many onshore/offshore developments.

Scanning: Elicit know-how from business users and proxies

Abstract and codify: Generate design documents

Diffuse: Send material to India for development

Absorb: Coders read **and understand** the material

Impact: Code returns for test

Note: The more I use this model the more I see it as a potential fractal component as it can be applied at many levels.

Ghoshal's 'organising framework'

Sources of competitive advantage

Strategic objectives	NATIONAL DIFFERENCES	SCALE ECONOMIES	SCOPE ECONOMIES
Achieving efficiency in current operations	Benefiting from differences in factor costs (e.g. wages and cost of capital)	Expanding and exploiting potential scale economies in each activity	Sharing of investments and costs across products, markets and businesses
Managing risks	Managing different kinds of risk arising from market or policy-induced changes in comparative advantages of different countries	Balancing scale with strategic and operational flexibility	Portfolio diversification of risks and creation of options and side-bets
Innovation, learning and adaptation	Learning from societal differences in organisational and managerial processes and systems	Benefiting from experience (cost reduction and innovation)	Shared learning across organisational components in different products, markets or businesses

(Ghoshal, 1987)

16

Ghosal's organizing framework is an ideal blueprint for companies wanting to leverage scope and scale from global operations while also increasing efficiency and minimizing risk on a firm foundation of learning and innovation. It was the key stone in the process.

Bear in mind that there are two value chains to be considered: Firstly is the client who is hoping to secure reduced costs through developer arbitrage and secondly is the offshore outfit that is looking to train its staff and learn new business processes. In short it takes the knowledge from one company and redeploys it on subsequent clients.

Developer arbitrage only works if the developers in different areas are of comparable quality. The main Indian players are so large that the quality of their staff varies. How do you ensure you get the level of expertise you require? Quizzes, brainbenching

Industrial age vs. knowledge age management

Table 2.2 From industrial age management of work to knowledge age management of work		
	Industrial age work (20th century)	Knowledge age work (21st century)
Values and technology	Machine tools	Information and communication technology (ICT)
Core resources	Capital/labour intensive	Knowledge intensive
Core values	Solidarity	Individualism/teamwork
Basis of order	Compliance	Commitment
Knowledge distribution	Knowledge fragmentation	Knowledge sharing
Goal (source of competitive advantage)	Efficient operations (exploitation)	New developments (exploration)
Role of innovation	Periodic	Continual
Management role	Boss	Coach
	Decision maker	Facilitator
	Source of information	Knowledge officer
Performance management approach	Supervision and monitoring	Facilitated opportunities
Organisational form	Pyramid	Inverted pyramid/spider's web/network
	Vertical communication	Lateral communication
	Centralised	Devolved
People management practices	Command and control	Empowerment
	Low priority to selection	Careful selection
	Division of labour	Shared knowledge
	Prioritising explicit knowledge	Valuing tacit knowledge
	Low skill/limited training	High skill/lifelong learning
	Knowledge 'silos'	Shared knowledge
	Training centres	Open learning

During the release retrospective it was heartening to hear the team say that we had done all the things in the right hand column. It proves that the "noolithic" era techniques are valid and produce human results. The difficulty is in applying them within fossilised work practices.

Project Pivots

Projects go through different phases – defining the problem, exploring and defining the solution and implementing the solution. Time constraints can squeeze the phases and over simplistic views of the process results in mistakes.

The process matrix is a project level implementation of the Cynefin.

Agile Method Matrix

Select a methodology depending on project parameters

Be prepared to shift from method to method as the project progresses

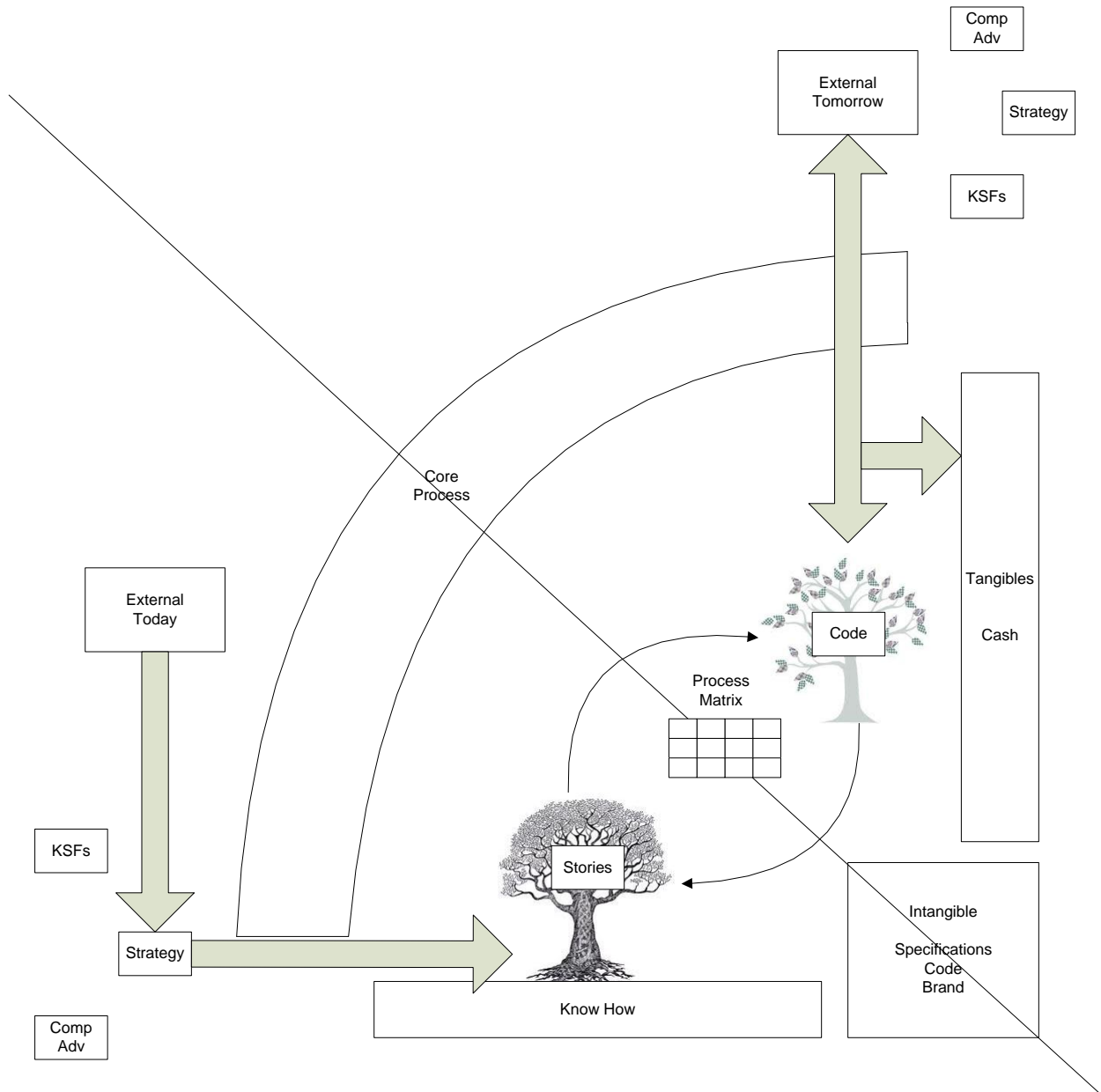
	New Practice	Emergent Practice	Good Practice	Best Practice
Complexity Level	Chaotic	Complex	Complicated	Simple
Design forces	Design Thinking Creative Problem Solving	Focus on User Empathy	Domain Driven Design	Fast Fix Service Driven
Methods	WickedAgile Buffalo SSM CATWOE	UCD BDD SERVQual	Value Chain Analysis UML/FDD TDD	JDI BAU Kanban

These can be considered as aspects of design thinking or creative problem solving.

A metaphorical deconstruction of the RBV showing the idealised conversion of know-how to code is presented over. Essentially it is the causal loops from earlier drawn in a friendlier way.

Biological metaphors are better than production line ones for knowledge creation.

The deconstructed representation of the Resource Based View of the organisation shows where the process matrix described on the previous page fits into a strategic model to produce sustained competitive advantage.



Linking the work being carried out in the project can be fed into higher level measurement systems through the house of value (HoV) reports. The HoV can easily be constructed by iterating through a Kaplan and Norton style Strategic map breakdown.