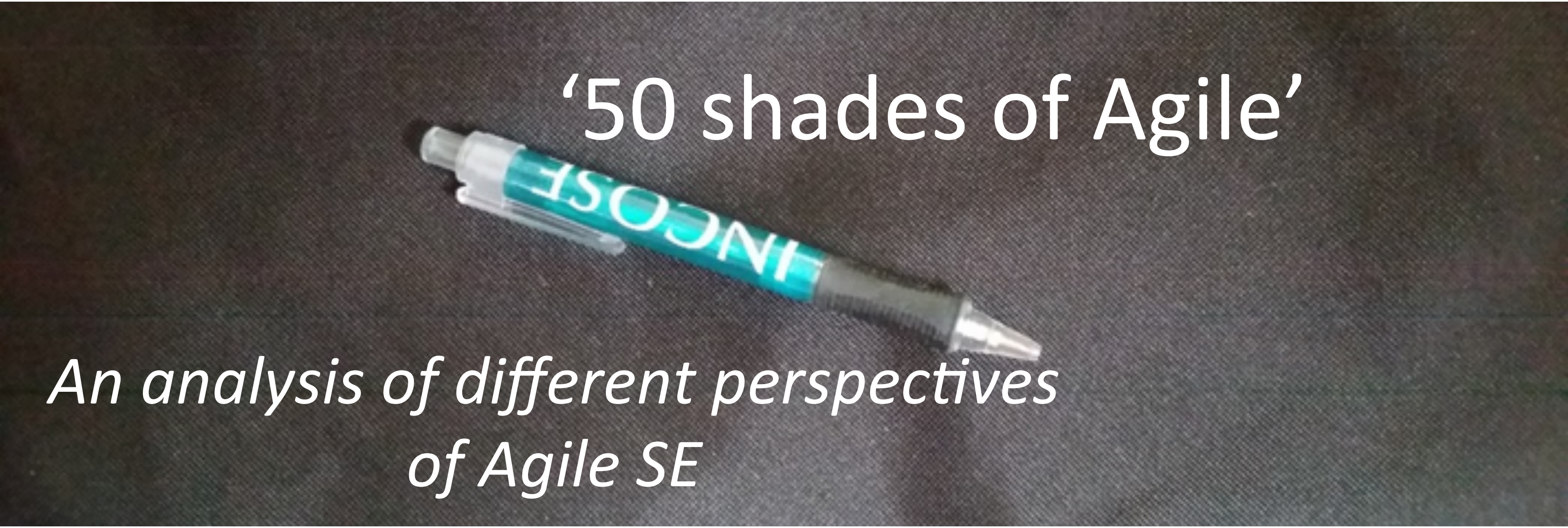
26th annual **INCOSE**
international symposium

**Edinburgh, UK**
July 18 - 21, 2016



# ‘50 shades of Agile’

*An analysis of different perspectives of Agile SE*

# The authors

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

+ Kirsty

de&s

UNIVERSITY OF BIRMINGHAM

IBM

LEONARDO

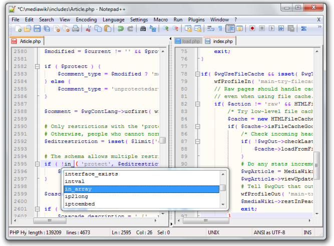
[dstl]

MBDA MISSILE SYSTEMS

This paper represents the views of the authors, and does not represent the position of the MOD, IBM, MBDA, Leonardo or the University of Birmingham.

# A long time ago in a galaxy far-far away

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016



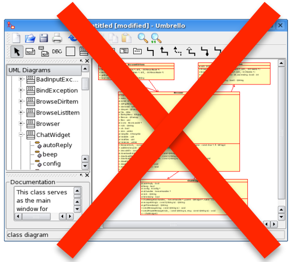
*C:\mediawiki\includes\Article.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ? X
Article.php
load.php
index.php

```
$modified = $current != '' && $pro
    
if ( $protect ) {
    $comment_type = $modified ? 'm
} else {
    $comment_type = 'unprotectedar
}

$comment = $wgContLang->ucfirst( w

# Only restrictions with the 'prot
# Otherwise, people who cannot nor
$editrestriction = isset( $limit['

# The schema allows multiple restr
if ( !in ( 'protect', $editrestric
    $
}

$casc

if (
    $cascade_description = ' ['
```

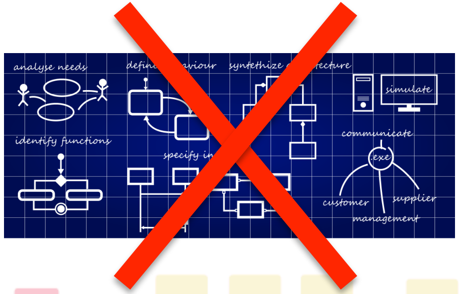
```
interface_exists
intval
in_array
ip2long
iptcembed
```

```
    exit;
}

if( $wgUseFileCache && isset( $wg
    wfProfileIn( 'main-try-fileca
    // Raw pages should handle cac
    // even when using file cache.
    if( $action != 'raw' && HTMLF
        /* Try low-level file cach
        $cache = new HTMLFileCache
        if( $cache->isFileCacheGo
            /* Check incoming hea
            if( !$wgOut->checkLast
                $cache->loadFromF
            }
            # Do any stats increme
            $wgArticle = MediaWik
            $wgArticle->viewUpdate
            # Tell $wgOut that out
            wfProfileOut( 'main-t
            $mediaWiki->restInPea
            exit;
```

PHP Hy length : 139209 lines : 4673 Ln : 2595 Col : 26 Sel : 0 UNIX ANSI as UTF-8 INS

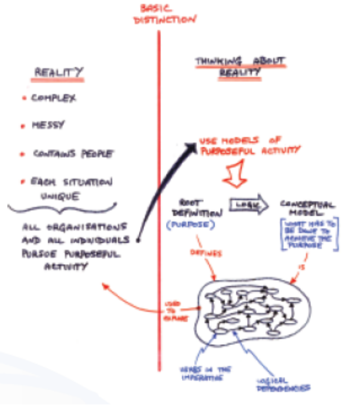


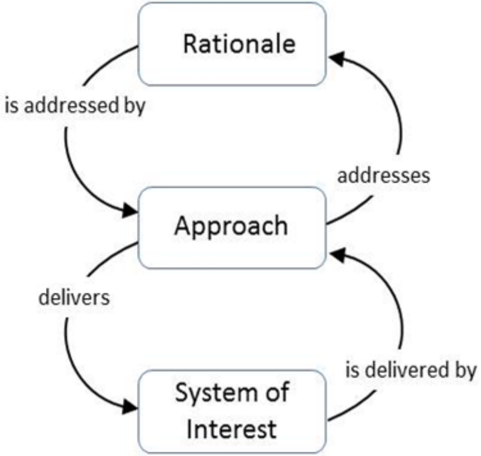
This class serves as the main window for

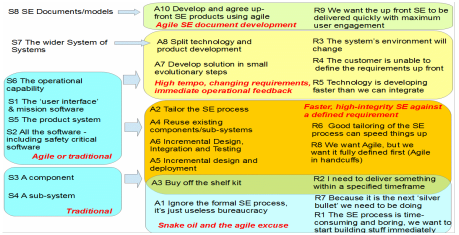


# But software is different from systems

# Approach taken in paper




Rationale
is addressed by
addresses
Approach
delivers
is delivered by
System of Interest


S8 SE Documents/models

A10 Develop and agree up-front SE products using agile

*Agile SE document development*

R9 We want the up front SE to be delivered quickly with maximum user engagement

S7 The wider System of Systems

A8 Split technology and product development

A7 Develop solution in small evolutionary steps

*High tempo, changing requirements, immediate operational feedback*

R3 The system's environment will change

R4 The customer is unable to define the requirements up front

R5 Technology is developing faster than we can integrate

S6 The operational capability

S1 The 'user interface' & mission software

S5 The product system

S2 All the software - including safety critical software

*Agile or traditional*

A2 Tailor the SE process

A4 Reuse existing components/sub-systems

A6 Incremental Design, Integration and Testing

A5 Incremental design and deployment

*Faster, high-integrity SE against a defined requirement*

R6 Good tailoring of the SE process can speed things up

R8 We want Agile, but we want it fully defined first (Agile in handcuffs)

S3 A component

S4 A sub-system

*Traditional*

A3 Buy off the shelf kit

R2 I need to deliver something within a specified timeframe

A1 Ignore the formal SE process, it's just useless bureaucracy

*Snake oil and the agile excuse*

R7 Because it is the next 'silver bullet' we need to be doing

R1 The SE process is time-consuming and boring, we want to start building stuff immediately

# Rationale for Agile

R1 The SE process is time-consuming and boring, we want to start building stuff immediately

R2 I need to deliver something within a specified timeframe

R3 The system's environment will change

R4 The customer is unable to define the requirements up front

R5 Technology is developing faster than we can integrate

R6 Good tailoring of the SE process can speed things up

R7 Because it is the next 'silver bullet' we need to be doing

R8 We want Agile, but we want it fully defined first (Agile in handcuffs)

R9 We want the up-front SE to be delivered quickly with maximum user engagement

# Systems we could develop using agile

S1 The ‘user interface’ & mission software

S2 All the software – including safety critical software

S3 A component

S4 A sub-system

S5 The product system

S6 The operational capability

S7 The wider System of Systems

S8 SE Documents/models

Image © Network Rail

# Approaches to agile

A1 Ignore the formal SE process, it's just useless bureaucracy

A2 Tailor the SE process

A3 Buy off the shelf kit

A4 Reuse existing components/sub-systems

A5 Incremental design and deployment

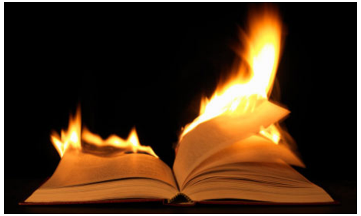
A6 Incremental design, integration and testing

A7 Develop solution in small evolutionary steps

A8 Split technology and product development

A9 Half-baked transition

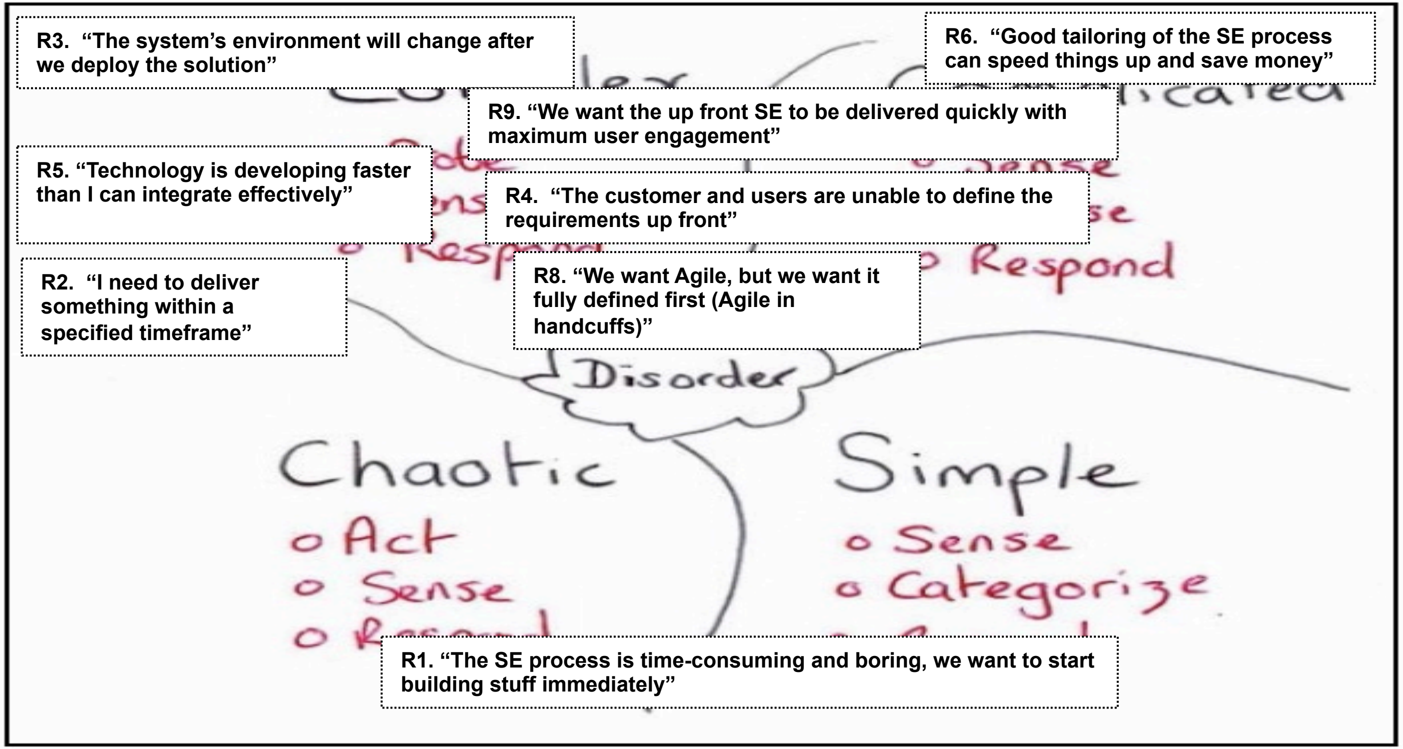
A10 Develop and agree up-front SE products using Agile



# Analysis

- We analysed the weltanschauungen against
  - Cynefin framework
  - Ring cycle
  - 12 agile principles
  - *Hitchins' 5 layer model*

# Cynefin



R3. “The system’s environment will change after we deploy the solution”

R6. “Good tailoring of the SE process can speed things up and save money”

R9. “We want the up front SE to be delivered quickly with maximum user engagement”

R5. “Technology is developing faster than I can integrate effectively”

R4. “The customer and users are unable to define the requirements up front”

R2. “I need to deliver something within a specified timeframe”

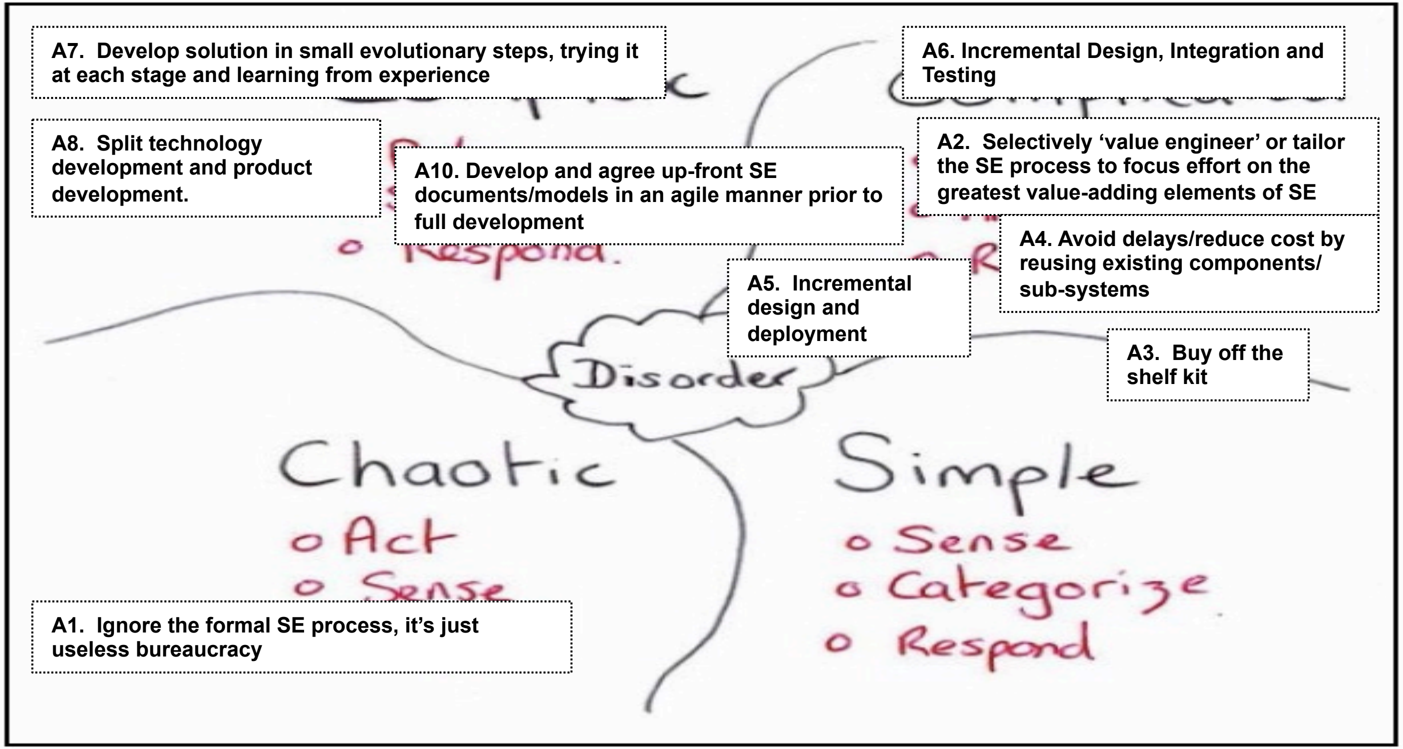
R8. “We want Agile, but we want it fully defined first (Agile in handcuffs)”

R1. “The SE process is time-consuming and boring, we want to start building stuff immediately”

# Cynefin

26th annual INCOSE international symposium

Edinburgh, UK
July 18 - 21, 2016



A7. Develop solution in small evolutionary steps, trying it at each stage and learning from experience

A6. Incremental Design, Integration and Testing

A8. Split technology development and product development.

A10. Develop and agree up-front SE documents/models in an agile manner prior to full development

A2. Selectively ‘value engineer’ or tailor the SE process to focus effort on the greatest value-adding elements of SE

A4. Avoid delays/reduce cost by reusing existing components/ sub-systems

A5. Incremental design and deployment

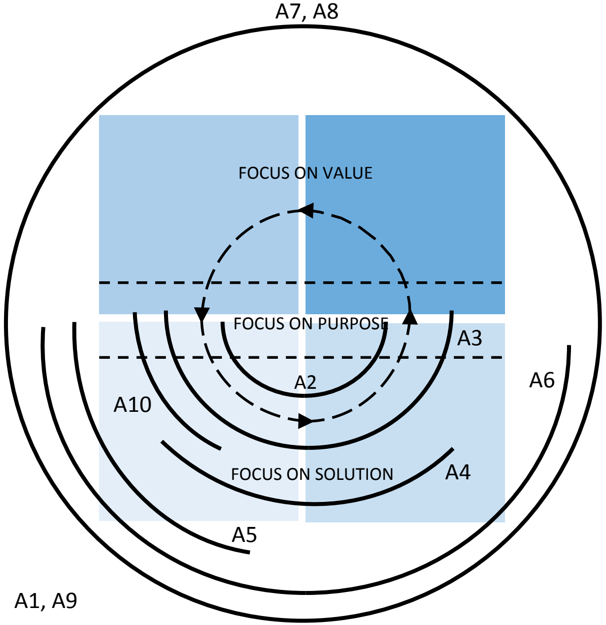
A3. Buy off the shelf kit

A1. Ignore the formal SE process, it’s just useless bureaucracy

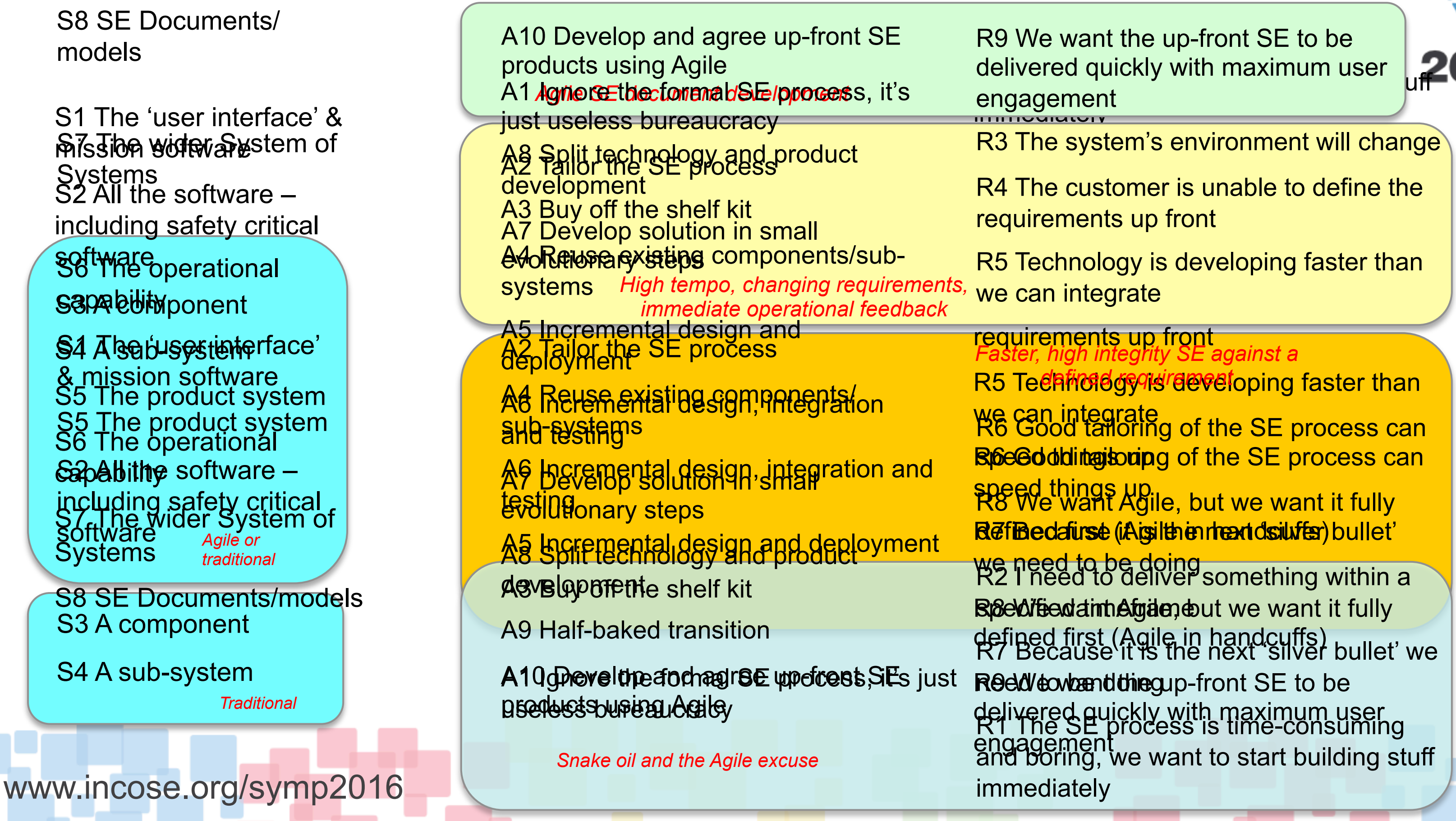
# Agile principles

| | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A10 |
|---|---|---|---|---|---|---|---|---|---|
| Early and continuous delivery | 🟥 | 🟥 | 🟥 | 🟥 | 🟨 | 🟩 | 🟩 | 🟩 | 🟨 |
| Welcome changing requirements | 🟨 | 🟥 | 🟥 | 🟥 | 🟨 | 🟩 | 🟩 | 🟩 | 🟥 |
| Deliver working software frequently | 🟨 | 🟥 | 🟥 | 🟥 | 🟨 | 🟨 | 🟩 | 🟩 | 🟥 |
| Work together daily | 🟥 | 🟥 | 🟥 | 🟨 | 🟨 | 🟥 | 🟨 | 🟨 | 🟨 |
| Motivated individuals | 🟩 | 🟥 | 🟥 | 🟥 | 🟨 | 🟥 | 🟨 | 🟨 | 🟨 |
| Face-to-face conversation. | 🟩 | 🟥 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟥 |
| Working software | 🟨 | 🟥 | 🟨 | 🟥 | 🟨 | 🟨 | 🟩 | 🟩 | 🟥 |
| Sustainable development | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 |
| Technical excellence and good design | 🟥 | 🟩 | 🟥 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 |
| Simplicity | 🟥 | 🟩 | 🟨 | 🟩 | 🟨 | 🟥 | 🟥 | 🟨 | 🟩 |
| Self-organizing teams. | 🟨 | 🟥 | 🟥 | 🟥 | 🟨 | 🟥 | 🟨 | 🟨 | 🟩 |
| Team reflects and adjusts behaviour | 🟥 | 🟥 | 🟥 | 🟨 | 🟩 | 🟥 | 🟩 | 🟨 | 🟩 |

# Ring cycle


A7, A8
FOCUS ON VALUE
FOCUS ON PURPOSE
A3
A2
A6
A10
FOCUS ON SOLUTION
A4
A5
A1, A9

# Summary of analysis



S8 SE Documents/ models

S1 The ‘user interface’ & mission software

S7 The wider System of Systems

S2 All the software – including safety critical software

S6 The operational capability

S3 A component

S1 The ‘user interface’ & mission software

S4 A sub-system

S5 The product system

S5 The product system

S6 The operational capability

S2 All the software – including safety critical software

S7 The wider System of Systems

*Agile or traditional*

S8 SE Documents/models

S3 A component

S4 A sub-system

*Traditional*

A10 Develop and agree up-front SE products using Agile

A1 Ignore the formal SE process, it’s just useless bureaucracy

*Agile SE document development*

R9 We want the up-front SE to be delivered quickly with maximum user engagement

uff

immediately

A8 Split technology and product development

A2 Tailor the SE process

A3 Buy off the shelf kit

A7 Develop solution in small evolutionary steps

A4 Reuse existing components/sub-systems

*High tempo, changing requirements, immediate operational feedback*

R3 The system’s environment will change

R4 The customer is unable to define the requirements up front

R5 Technology is developing faster than we can integrate

A5 Incremental design and deployment

requirements up front

A2 Tailor the SE process

*Faster, high integrity SE against a defined requirement*

A4 Reuse existing components/ sub-systems

A6 Incremental design, integration and testing

R5 Technology is developing faster than we can integrate

R6 Good tailoring of the SE process can speed things up

R6 Good tailoring of the SE process can speed things up

A6 Incremental design, integration and testing

A7 Develop solution in small evolutionary steps

R8 We want Agile, but we want it fully defined first (Agile in handcuffs)

R7 Because it is the next ‘silver bullet’ we need to be doing

A5 Incremental design and deployment

A8 Split technology and product development

A3 Buy off the shelf kit

R2 I need to deliver something within a specified timeframe

R8 We want Agile, but we want it fully defined first (Agile in handcuffs)

A9 Half-baked transition

R7 Because it is the next ‘silver bullet’ we need to be doing

A10 Develop and agree up-front SE products using Agile

A1 Ignore the formal SE process, it’s just useless bureaucracy

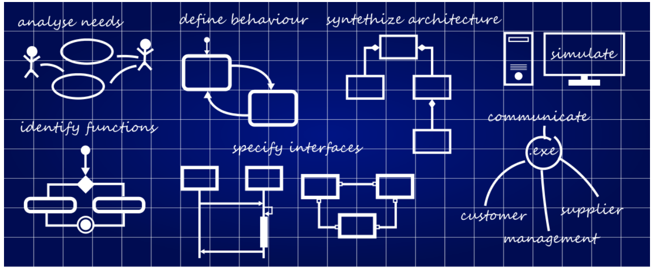
R9 We want the up-front SE to be delivered quickly with maximum user engagement

R1 The SE process is time-consuming and boring, we want to start building stuff immediately

*Snake oil and the Agile excuse*

# Four shades of agile?

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016



SNAKE OIL LINIMENT

THE STRONGEST AND BEST LINIMENT KNOWN FOR PAIN AND LAMENESS.

USED EXTERNAL ONLY.

FOR
RHEUMATISM
NEURALGIA
SCIATICA
LAME BACK
LUMBAGO
CONTRACTED CORDS
TOOTHACHE
SPRAINS
SWELLINGS
ETC.

CLARK STANLEY'S SNAKE OIL LINIMENT

TRADE MARK REGISTERED

—FOR—
FROST BITES
CHILL BLAINS
BRUISES
SORE THROAT
BITES OF ANIMALS INSECTS AND REPTILES.

GOOD FOR MAN AND BEAST

IT GIVES IMMEDIATE RELIEF.

IS GOOD FOR EVERYTHING A LINIMENT OUGHT TO BE GOOD FOR

Manufactured by
CLARK STANLEY
Snake Oil Liniment
Company
Providence, R. I.

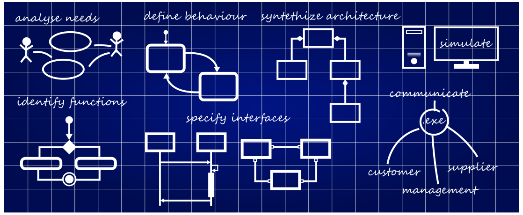
Clark Stanley's Snake Oil Liniment

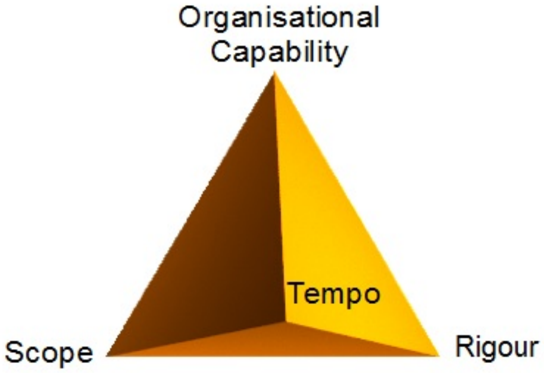
Is for sale by all druggists. If your druggist fails to have it tell him he can get it for you from any wholesale druggists or it will be sent to you to any part of the United States or Canada upon the receipt of fifty cents in stamps by addressing the

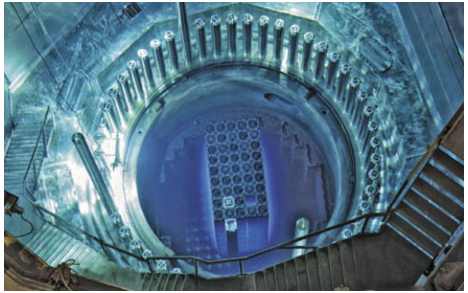
Clark Stanley Snake Oil Liniment Co.

PROVIDENCE, R. I.

# Where does agile sit?







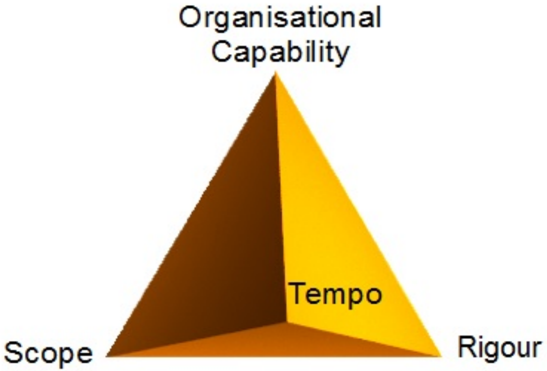
# Perceptions and misunderstandings

- No common understanding
- Risk it becomes a meaningless label
- Move to Agile difficult because we don’t understand why low tempo SE works!
- Operational use is a critical element of the Agile approach.
- Agile approaches can be used to develop SE documents or models.
- Agile rigour is different to conventional rigour

# Agile SE is different

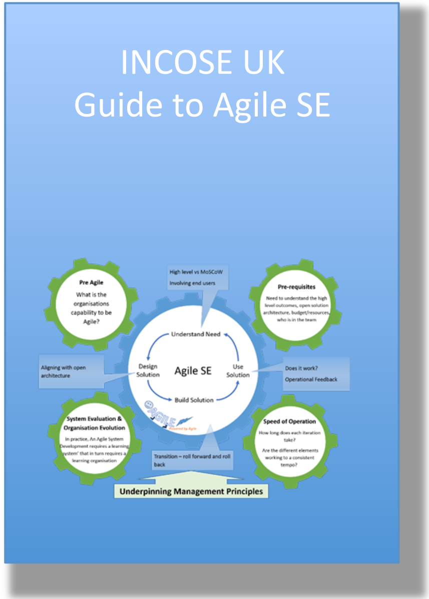
- Agile software techniques not always applicable to SE.
- Agile SE is different to Agile Software Engineering.
- Conventional and agile SE are in different part of the trade space
- Conventional SE manages risks by managing them, agile SE by avoiding them

# Different SE for different places



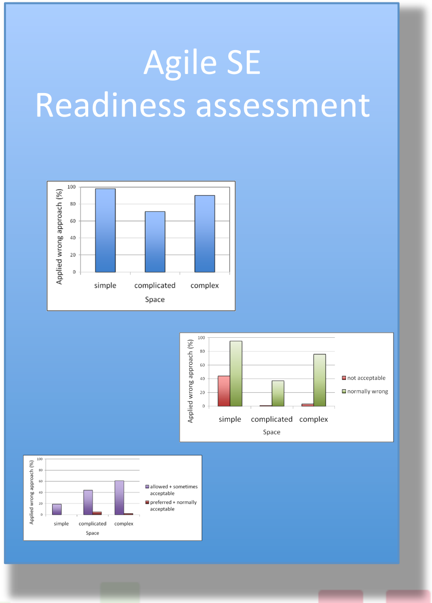
- The Agile SE and conventional SE paradigms are fundamentally different
- Conventional works well in complicated space
- Agile works well in complex spaces
- Conventional SE works best when assurance, design, manufacture and installation rework costs are significant
- The more effective the organisation the better able to do agile and conventional SE

# Future work



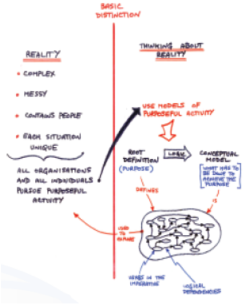
INCOSE UK
Guide to Agile SE

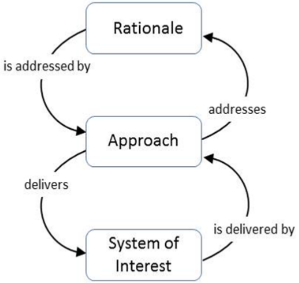
Pre Agile
What is the organisations capability to be Agile?
High level vs MoSCoW
Involving end users
Pre-requisites
Need to understand the high level outcomes, open solution architecture, budget/resources, who is in the team
Understand Need
Aligning with open architecture
Design Solution
Agile SE
Use Solution
Does it work?
Operational Feedback
Build Solution
AGILE
Powered by Agile
System Evaluation & Organisation Evolution
In practice, An Agile System Development requires a learning system' that in turn requires a learning organisation
Transition – roll forward and roll back
Speed of Operation
How long does each iteration take?
Are the different elements working to a consistent tempo?
Underpinning Management Principles

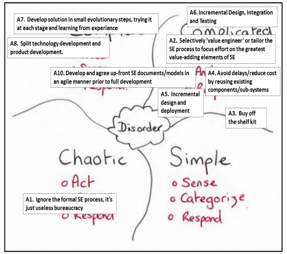


Agile SE
Readiness assessment

# Summary







A7. Develop solution in small evolutionary steps, trying it at each stage and learning from experience

A6. Incremental Design, Integration and Testing

A8. Split technology development and product development.

A2. Selectively 'value engineer' or tailor the SE process to focus effort on the greatest value-adding elements of SE

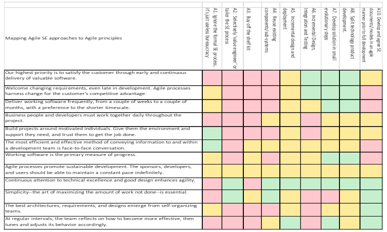
A10. Develop and agree up-front SE documents/models in an agile manner prior to full development

A4. Avoid delays/reduce cost by reusing existing components/sub-systems

A5. Incremental design and deployment

A3. Buy off the shelf kit

A1. Ignore the formal SE process, it's just useless bureaucracy



| Mapping Agile SE approaches to Agile principles | A1. Ignore the formal SE process, it's just useless bureaucracy | A2. Selectively 'value engineer' or tailor the SE process | A3. Buy off the shelf kit | A4. Reuse existing components/sub-systems | A5. Incremental design and deployment | A6. Incremental Design, Integration and Testing | A7. Develop solution in small evolutionary steps | A8. Split technology product development. | A10. Develop and agree SE documents/models in an agile manner prior to full development |
|---|---|---|---|---|---|---|---|---|---|
| Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. | 🟥 | 🟥 | 🟥 | 🟥 | 🟨 | 🟩 | 🟩 | 🟩 | 🟨 |
| Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. | 🟨 | 🟥 | 🟥 | 🟥 | 🟨 | 🟩 | 🟩 | 🟩 | 🟥 |
| Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. | 🟨 | 🟥 | 🟥 | 🟥 | 🟨 | 🟨 | 🟩 | 🟩 | 🟥 |
| Business people and developers must work together daily throughout the project. | 🟥 | 🟥 | 🟥 | 🟨 | 🟨 | 🟥 | 🟨 | 🟨 | 🟨 |
| Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. | 🟩 | 🟥 | 🟥 | 🟥 | 🟨 | 🟥 | 🟨 | 🟨 | 🟨 |
| The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. | 🟩 | 🟥 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟥 |
| Working software is the primary measure of progress. | 🟨 | 🟥 | 🟨 | 🟥 | 🟨 | 🟨 | 🟩 | 🟩 | 🟥 |
| Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 |
| Continuous attention to technical excellence and good design enhances agility. | 🟥 | 🟩 | 🟥 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 |
| Simplicity--the art of maximizing the amount of work not done--is essential. | 🟥 | 🟩 | 🟨 | 🟩 | 🟨 | 🟥 | 🟥 | 🟨 | 🟩 |
| The best architectures, requirements, and designs emerge from self-organizing teams. | 🟨 | 🟥 | 🟥 | 🟥 | 🟨 | 🟥 | 🟨 | 🟨 | 🟩 |
| At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. | 🟥 | 🟥 | 🟥 | 🟨 | 🟩 | 🟥 | 🟩 | 🟨 | 🟩 |

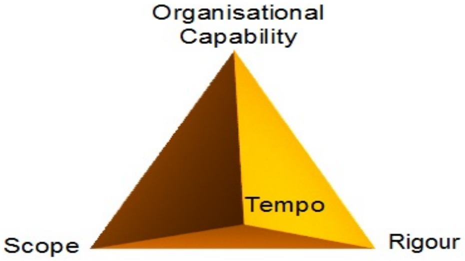
SNAKE OIL LINIMENT
THE STRONGEST AND BEST LINIMENT KNOWN FOR PAIN AND LAMENESS.
USED EXTERNAL ONLY
FOR RHEUMATISM NEURALGIA SCIATICA LAME BACK LUMBAGO CONTRACTED CORDS TOOTHACHE SPRAINS SWELLINGS ETC.
CLARK STANLEY'S
SNAKE OIL LINIMENT
TRADE MARK REGISTERED
FOR FROST BITES CHILL BLAINS BRUISES SORE THROAT BITES OF ANIMALS INSECTS AND REPTILES.
GOOD FOR MAN AND BEAST
IT GIVES IMMEDIATE RELIEF.
IS GOOD FOR EVERYTHING A LINIMENT OUGHT TO BE GOOD FOR
Manufactured by CLARK STANLEY Snake Oil Liniment Company Providence, R. I.

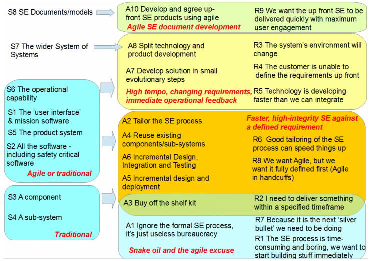
Clark Stanley's Snake Oil Liniment

Is for sale by all druggists. If your druggist fails to have it tell him he can get it for you from any wholesale druggists or it will be sent to you to any part of the United States or Canada upon the receipt of fifty cents in stamps by addressing the

Clark Stanley Snake Oil Liniment Co.

PROVIDENCE, R. I.





S8 SE Documents/models

S7 The wider System of Systems

S6 The operational capability

S1 The 'user interface' & mission software

S5 The product system

S2 All the software - including safety critical software

*Agile or traditional*

S3 A component

S4 A sub-system

*Traditional*

A10 Develop and agree up-front SE products using agile

*Agile SE document development*

R9 We want the up front SE to be delivered quickly with maximum user engagement

A8 Split technology and product development

A7 Develop solution in small evolutionary steps

*High tempo, changing requirements, immediate operational feedback*

R3 The system's environment will change

R4 The customer is unable to define the requirements up front

R5 Technology is developing faster than we can integrate

A2 Tailor the SE process

A4 Reuse existing components/sub-systems

A6 Incremental Design, Integration and Testing

A5 Incremental design and deployment

*Faster, high-integrity SE against a defined requirement*

R6 Good tailoring of the SE process can speed things up

R8 We want Agile, but we want it fully defined first (Agile in handcuffs)

A3 Buy off the shelf kit

R2 I need to deliver something within a specified timeframe

A1 Ignore the formal SE process, it's just useless bureaucracy

*Snake oil and the agile excuse*

R7 Because it is the next 'silver bullet' we need to be doing

R1 The SE process is time-consuming and boring, we want to start building stuff immediately