

An outcome based approach to software process improvement

BCS SPIN SG, 8 September 2009

C. C. Shelley
OXFORD SOFTWARE ENGINEERING Ltd
9 Spinners Court, 53 West End,
Witney,
Oxfordshire
OX28 1NH
www.osel.co.uk
info@osel.co.uk
Tel. +44 (0) 1993 700878

Aims:

To look at the current state of software process improvement...

...and propose simple approaches to improve SPI outcomes

Contents

- **The current state of software process improvement (SPI)
focussing on CMMI...
...but not unique to CMMI**
- **Origins and development of SPI and CMMI
or - How we got to where we are**
- **What are the issues we face now?**
- **How do we deal with them?
...some solutions**

Current state of SPI and CMMI...

- **Business continues to seek better software development ability**
 - **Faster delivery**
 - **Increased responsiveness**
 - **Reduced costs**
 - **Better quality**
 - **etc...**
 - **And the ability to demonstrate (and market) this**
- **CMMI is a model for s/w practice and s/w process improvement (SPI) that offers to deliver (or more accurately facilitate) this**
- **It is very influential and has global recognition**
- **It is supported by numerous products, services and suppliers**
- **And continues to undergo development and expansion**

...Current state of SPI and CMMI

- **But**
 - **There are concerns**
 - **high failure rate**
 - **low ROI compared to other ‘treatments’**
 - **systemic problem with delivery of CMMI services**
 - **CMMI and SPI are being challenged and overtaken by other approaches**

- **Why?**
 - **The model may be over extended?**

 - **SPI, as practiced now, is distorted and is frequently failing to deliver tangible business benefits**

Origins and History...

- CMMI was preceded by 'A method for assessing the software engineering capability of contractors' (CMU/SEI-87-TR-23) and later CMM
- These were developed for DoD as a *risk management tool* use to *select suppliers*
 - 'harsh' scoring system – organization graded as one of five levels, most of us are at base level.
 - This scoring persists in their successors
- Suppliers responded by working to meet CMM requirements
 - Note: The first use by s/w contractors was *not* SPI. It was 'conformance to std.' to enable suppliers to bid for work - but these contractors were engineering organizations that routinely use standards and were used to dealing with standard conformance issues
- CMM use expanded and used by others as a tool for Software Process Improvement (SPI)
 - With two primary uses
 - Risk management and compliance (mostly defence organizations at first) – using SCEs (audit like evaluations)
 - A framework for SPI – it is assumed that *high* maturity is better and *increasing* maturity lead to better performance – using CBA IPIs (assessments)

...Origins and History

- **(CMM was not the only approach to SPI**
 - SPI was already in place in software industry (mostly large US corporations)
 - CMM organized and popularized SPI
 - CMM becomes synonymous with SPI)
- **Increasing expansion of CMM services and customers**
 - Across industry sectors: spills out from defence and engineering to commercial, financial
 - World wide: and from US to Europe and then globally
 - And other CMMs developed (construction SPICE)
- **'Me to' models developed**
 - Bootstrap, SPICE (ISO 15504), Trillium, in house versions
- **CMMI developed**
 - Expands scope to systems and includes other aspects (suppliers and integration)
 - Adopts the continuous representation (which is largely ignored)
 - Increasing numbers of service providers, users and customers
 - Increasing numbers of courses and qualifications offered by SEI and other providers

Current state of SPI and CMMI

- Which brings us up to date. The good news is:
 - CMMI has global recognition and use as a model for s/w practice and improvement
 - Model(s) are developing and expanding – v. 1.2 currently – revision soon
 - CMMI compliance is required by many software customers
 - Introducing good practice to places it may not otherwise not have been

The bad news is:

- CMMI compliance is required by many software customers
 - Introduced to places where it is not understood or wanted, and causes difficulties
- Increasing concern with ROI and high failure rate
- Real SPI is becoming marginalized - improvement can become a 'tick box' exercise
- Unquestioning belief that model compliance automatically means better (and consequent loss of business trust in the model when this is found not to be so)

Is it just CMMI?...

- **The models still retain much of value that can be used in many software and systems environments**
- **The development and use of CMM and CMMI has a rich (if hidden) history, and many lessons learned**
- **Other approaches to better software development have gone through similar maturation, and more recent ones appear to be following the same path**
- **The problem is not in the model, but in how it is being promoted and used**

... So, in short - No

What are the issues we face now?...

- SPI is *risky*
 - It can be very expensive and time consuming, and unattractive to technical staff and managers, and deliver a poor return on investment ...

...What are the issues we face now?...

- **...Why?**
 - **Frequent misunderstanding, model misuse, and poor SPI practice**
 - **Models are (too?)complex, subtle, and incomplete**
 - **Treated as a design, rather than (incomplete) spec**
 - **Promoted as universally applicable, and with unrealistic benefits**
 - **Presumed that increased maturity automatically brings better capability**
 - **Compliance delivers performance**
 - **CMMI/SPI work often treated as a (big, expensive) project with synthetic, unrealistic and incomplete project objectives – e.g. ML3 in 2 years (& year on year productivity improvements)**
 - **Poor mapping of model to organization**
 - **or, more usually, organization to model**
 - **Assessments are now expensive, cumbersome ‘high stakes’ audits**
 - **Watch out for ‘gap analyses’, organizations ‘freeze’ to a maintain compliance**
 - **CMMI and SPI ‘specialists’ in often in ambiguous position**
 - **Assessor role has changed**
 - **SPI and process specialists often remote from the effect (good or bad) of SPI work**

...What to do about it?

- The models do contain many valuable features and ideas
 - they need to be recognized and *used*.
- Recognize and understand the risks
 - and *plan* appropriately and *act* accordingly

How do we deal with the issues? Some solutions...

Be realistic: we must acknowledge the real world situation:

- Many believe that model compliance and high maturity automatically delivers (unspecified) improved development performance – and have made major investments in that belief
- So ‘Big SPI’ projects will continue, with CMMI (and other models) imperfectly understood or interpreted

But real SPI is different:

- Different values and motivation
 - Business (objectives) first, model (compliance) second – always, the model should be *used*, not obeyed
- Different structure
 - Continuous, small changes, fast, accountable, exploratory, evidence based, learning, measured, honest

Real SPI often has to work within ‘Big SPI’ environment:

- Difficult, but...
- ...SPI must help those doing the work do a better job – *this is the only and proper purpose of SPI*
...and this reduces risks and timescales of ‘Big SPI’
- Use ‘Big SPI’ as an opportunity to introduce and do real SPI
- Big SPI and real SPI are not mutually exclusive – but co-existence can be difficult

...Solutions...

- **These solutions direct software process improvement away from unrealistic belief in models and unthinking compliance to identifying and solving software problems that get in the way of business objectives**
- **They are not intended a prescriptive, end-to-end methodology, but to characterize genuine SPI**

...leading to early and ongoing and compounding beneficial outcomes – directed to the business goals...

...engaged software professionals...

...and managers...

...in the development of a genuinely better software development capability...

...and earlier achievement of robust and demonstrable capability.

...Solutions...

1. Identify the real business objectives...

– How?...

- **It can take time to convince senior management this is needed**
- **Disturbing the belief that ‘compliance = better’ can take time and be ‘exciting’**
- **Model compliance may be one (or ‘the’, ‘essential’, ‘non negotiable’) objective**
 - **but is more compelling - and achievable - if translated into business benefits**
 - **ask why ML3 is required...**
 - **...tactfully**
 - **try asking about ‘before’ and ‘after’**

...Solutions...

1. ...Identify the real business objectives

...How?

- Find out who really cares about the business objectives**
- Try asking management to prioritize ‘faster, cheaper, better’**
- Use Post Implementation Reviews or Retrospectives (if performed) to reveal patterns of stresses - and by implication the business drivers)**
 - (and if not performed get them started, now)**
- Engage management in drafting software policies – *what* they want**
 - (wisdom from CMM carried on into CMMI)**
 - Share (for comment, revise, publish and maintain) policies**
 - Policies should be very *very* clear, specific and compelling statements of what is wanted**
 - Can be difficult**
 - Use objective outsiders to help find the business drivers and values**

...Solutions...

2. Identify problems getting in the way of the business objectives

How?

- **PIRs or Retrospectives again – you do have them working now?**
- **Conduct an investigation or appraisal**
- **Find owners for the problems – who suffers from the problem and wants it fixed?**
- **Group problems with the appropriate objectives to provide scope and traceability, to answer the question ‘Why are we solving this problem?’ when it is asked later**
- **Discard problems that cannot map to business objectives/goals**

...Solutions...

3. Fix Problems

How?

- **Owners work to find and implement many small manageable solutions**
 - **Explore software models for solutions – *use* the models**
 - **Envision what it will be like after the problem is fixed**
 - **Characterize ‘before’ and ‘after’ (and quantify the difference if you can)**
 - **hint: you can**
 - **Develop a simple action plan**
 - **Take small (fast, exploratory, timeboxed) steps – try it and see, failure is acceptable, and can be valuable**
 - **Evaluate the effects**
 - **Work collaboratively, share knowledge**

...Solutions...

4. Don't organize the SPI work as a project -

Why?

- **Projects are prone to project management, costed, budgeted, scheduled, resource constrained*, converge on an outcome (job done), compete with other projects, and need to appear successful**
- **SPI is systemic, collaborative, inclusive, exploratory, ongoing, open ended (capability on), experimental, continuous, rooted in reality (perception is *not* reality)**
- **SPI does need to be controlled and directed – but as part of the organizational 'business as usual', integrated in to day to day work**
- **Consider other work management models, perhaps a programme setting up PATs, managing change requests, or iterations, or improvements cycles...**

...Solutions...

5. Use a framework for problem solving and delivering improvements

Why?

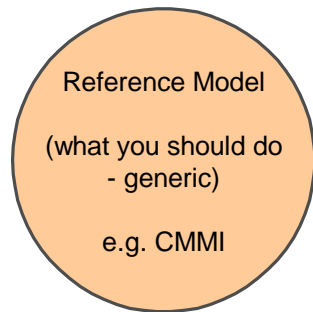
- **A framework for problem solving has the same benefits as a software lifecycle:**
 - **Familiar way of working**
 - **Comfort and safety for SPI workers - can be honest, removes blame from failure**
 - **Easier to plan and manage**
- **Examples:**
 - **your change control procedure (which may, on occasion, trigger small SPI projects)**
 - **Walter Shewhart's PDCA: Plan Do Check (or Study) Act,**
 - **The Six Sigma DMAIC - Define Measure Analyse Improve/Implement, Control (or maybe DFSS)**
 - **OSEL's TCM - www.osel.co.uk/rpi/c3.pdf**
 - **...or build one yourself**
- **Caveat – frameworks don't solve problems – people do, if they're allowed to**

...Solutions...

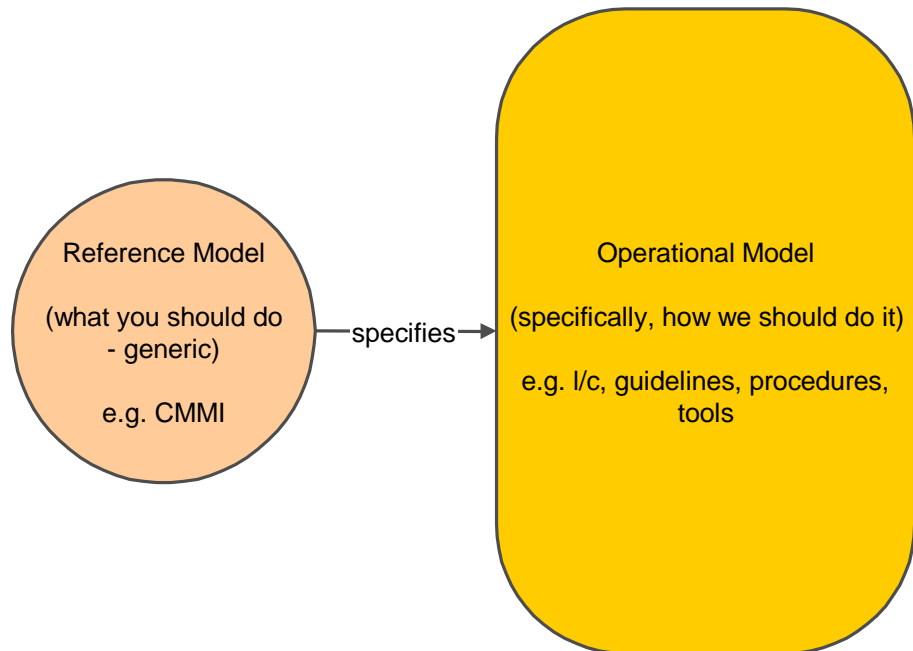
6. Use the models – understand them - *really* understand them - and who's in charge

How?

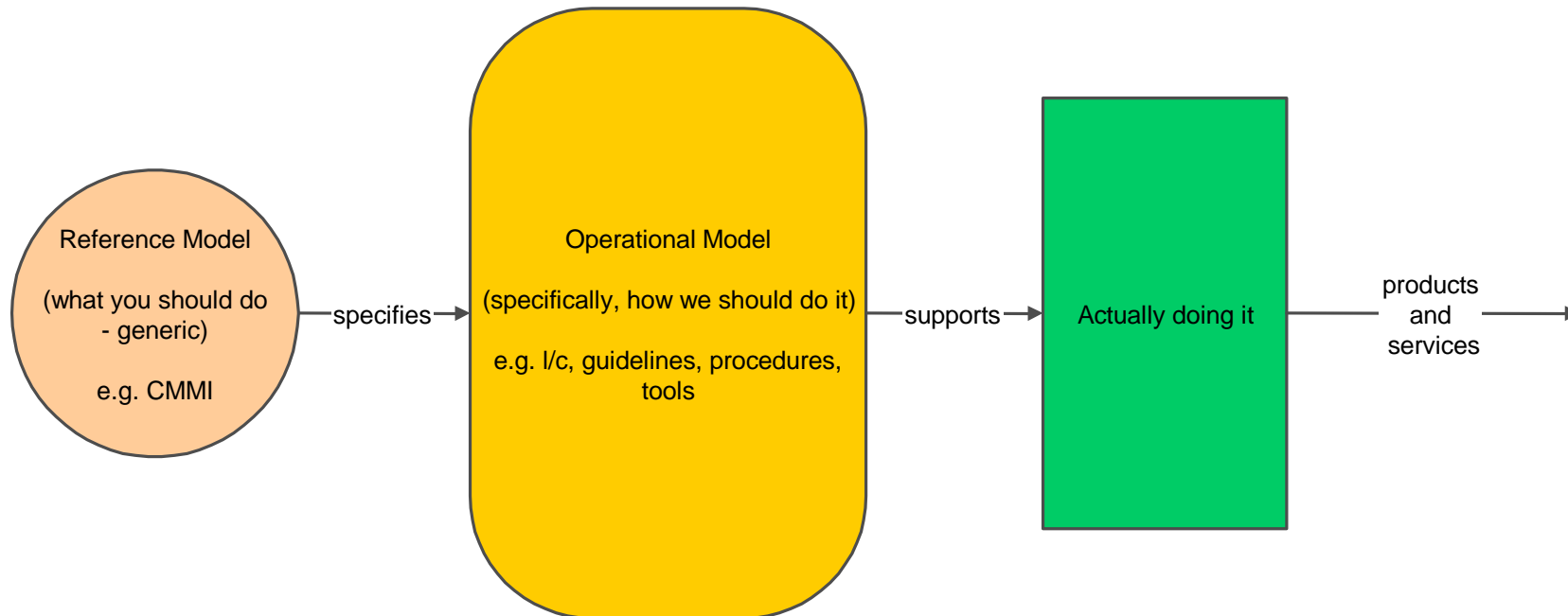
- Models contain much wisdom but are incomplete, imperfect - and dumb
- Using them as a design or *the sole* judge of goodness is a mistake.
- Use them as a guide and source of ideas and inspiration.
- If compliance is essential understand the model and how it maps onto your organization (*not visa versa*). Be prepared to explain and defend your ways of working from unthinking audits...
- ...your operational model is more important than someone else's reference model.



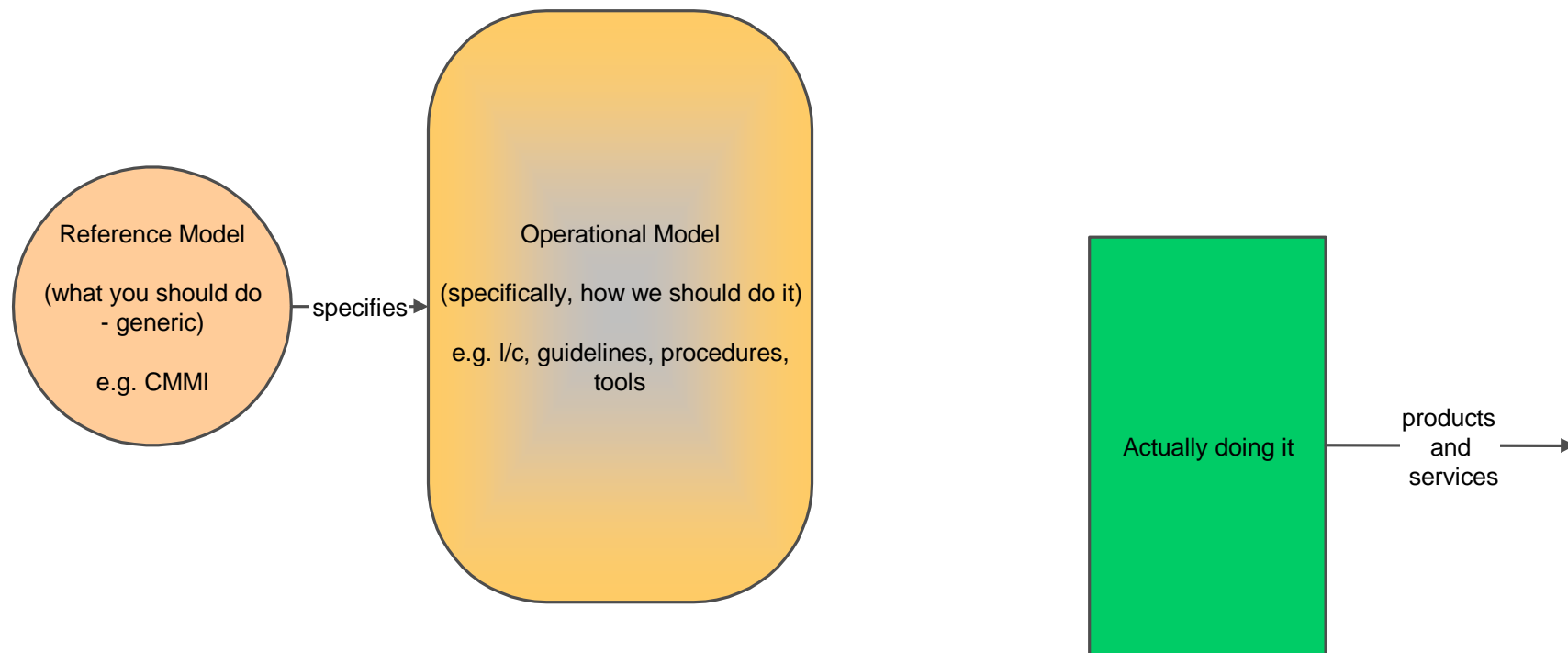
The software models and software practice



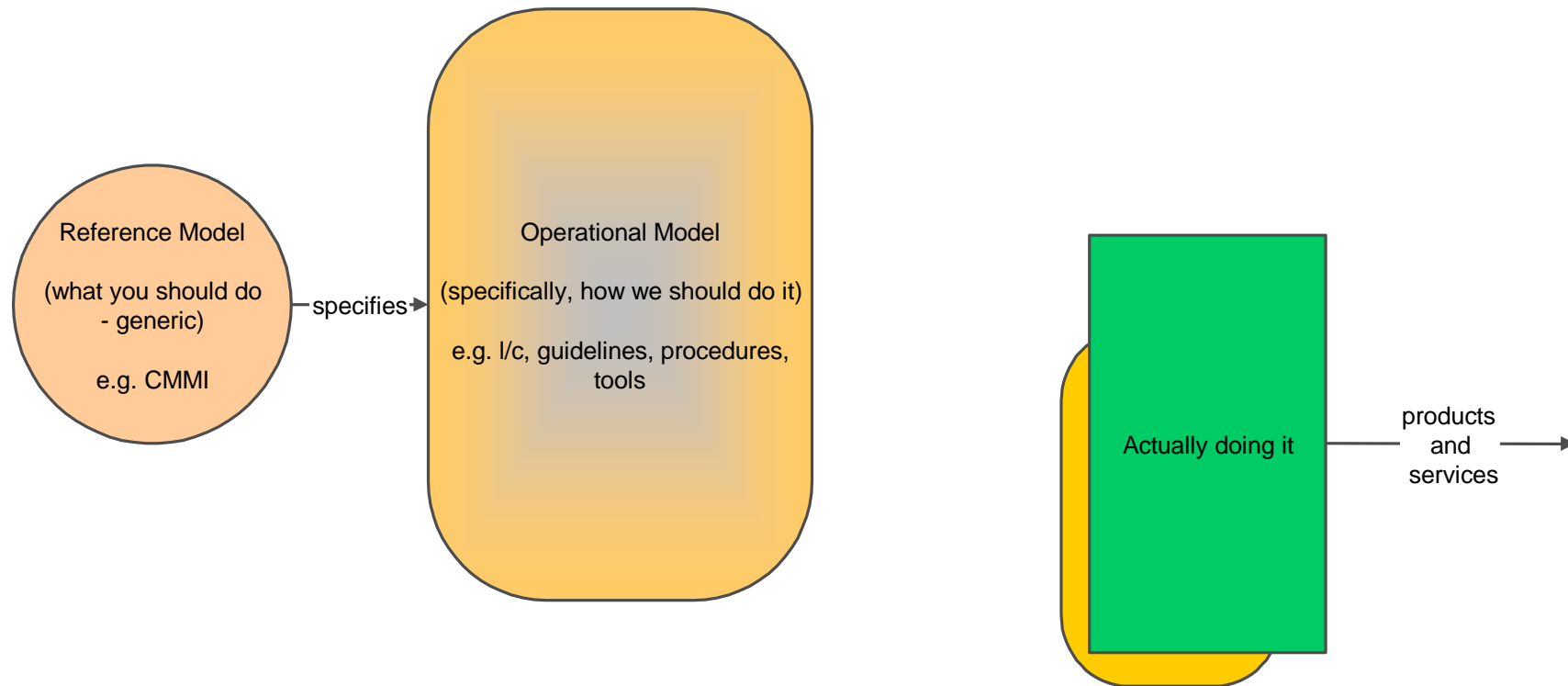
The software models and software practice



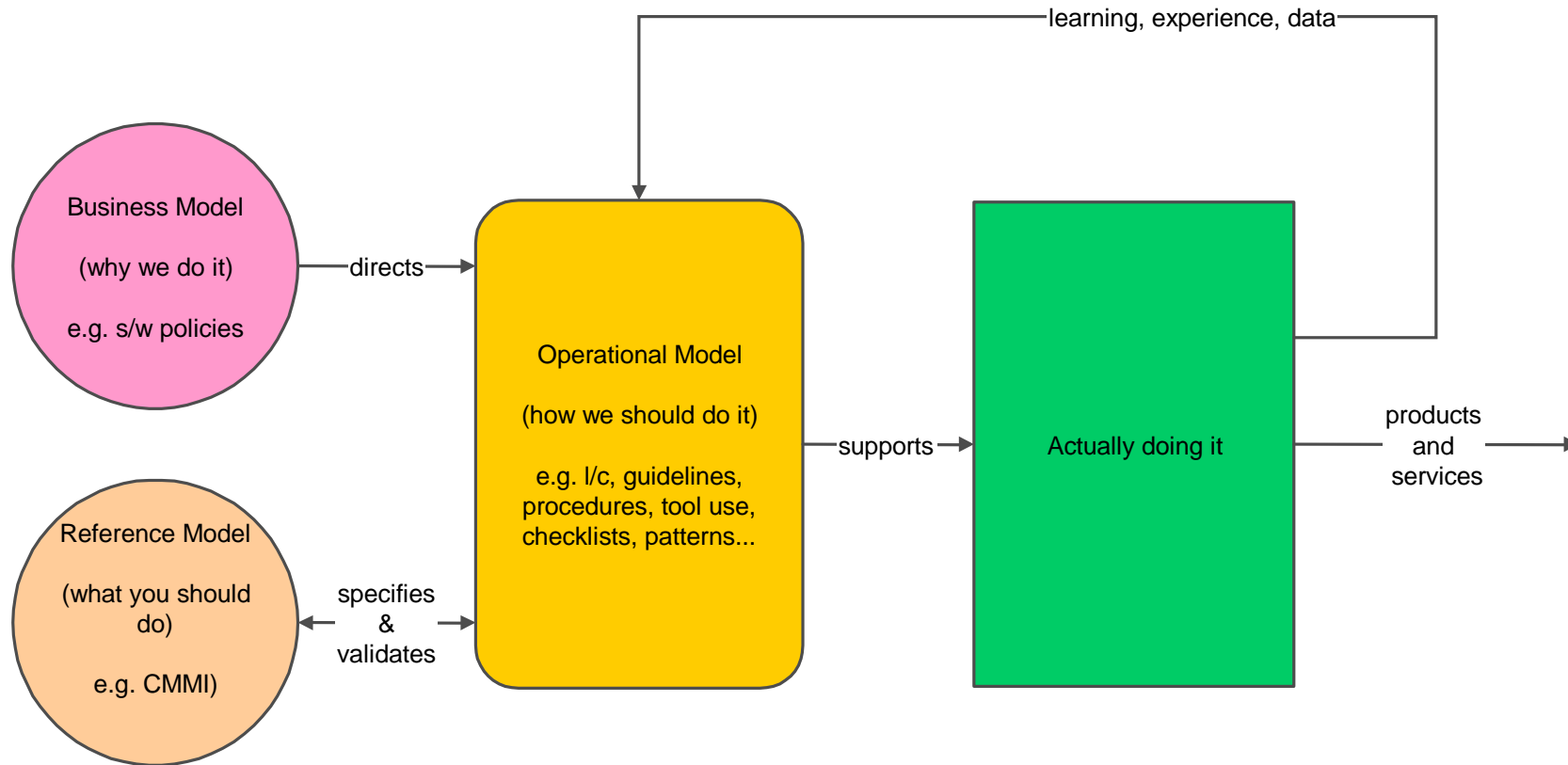
The software models and software practice



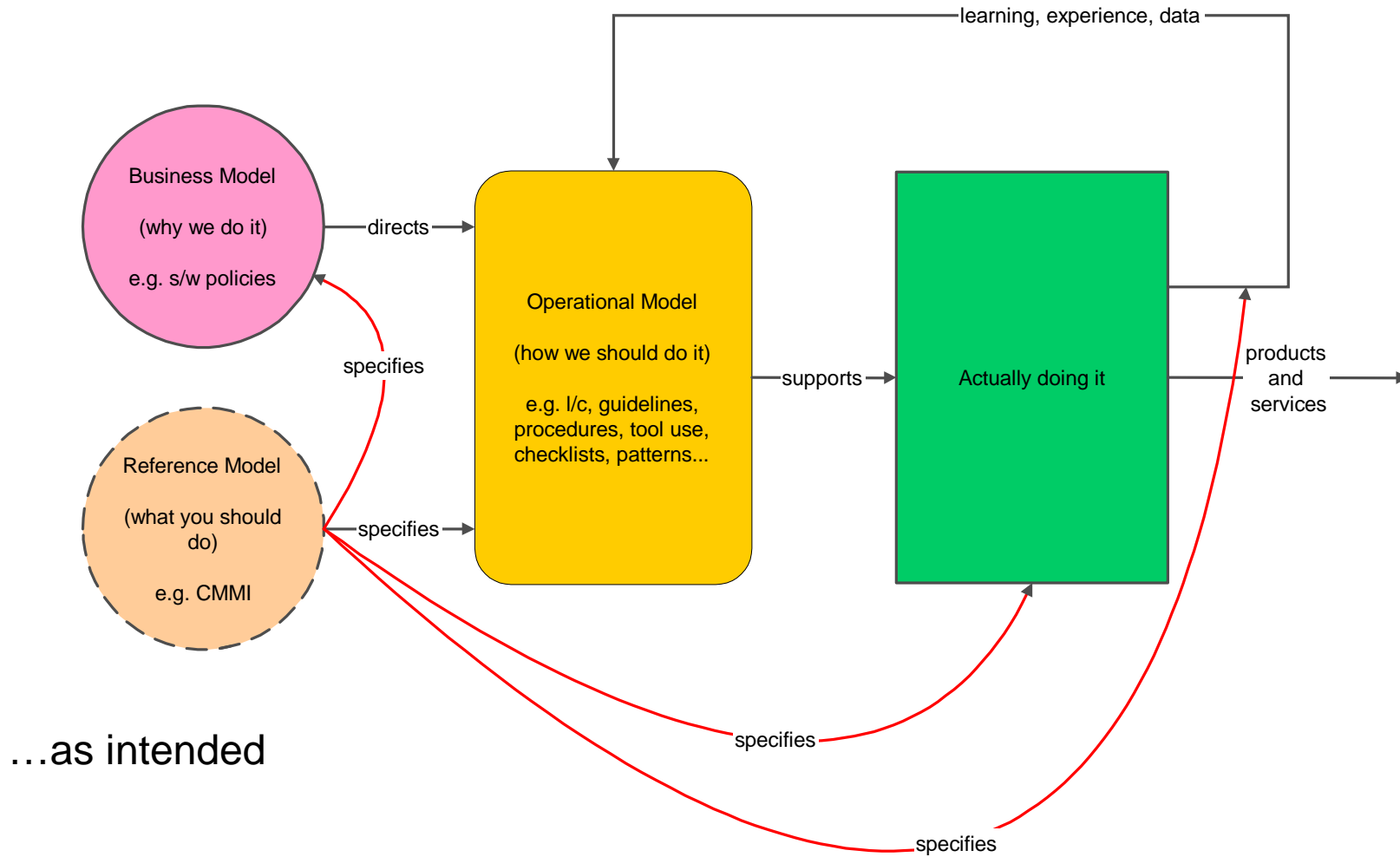
...focussed on compliance – limited value, shelfware, disappointed people



...focussed on compliance – limited value, shelfware, disappointed people



...focussed on business, first, and then compliance – a living system...



...Solutions...

7. Measuring Progress

Why?

- **Visibility & Control**
 - **How are we doing?**
 - **What works, what doesn't**

How?

- **Goal Question Metric**
 - **Tactical (applied to each change or improvement) and strategic (business performance)**
 - **Ms: e.g. Fixes, failures, benefits, rate of change**
 - **Progress – yes, Schedule - no**
 - **And elements of model satisfied (but only in conjunction with other measures – not on their own) Note: model is mapped to business – not visa versa**

...Solutions...

8. Keep going

Why?

- Things are getting better – why stop?**

Envoi:

For your consideration and debate: The Ten Rules of SPI...

...SPI must help those doing the work do a better job, therefore:

- 1. Software process improvements are owned by those that do the work**
- 2. Concentrate on fixing real problems getting in the way of business goals. If you aren't have a d****d good reason.**
- 3. Require rapid feedback (results) on the effect of your changes: solve lots of small problems fast...**
- 4. ...and evaluate (measure and analyse) them, and then act on them.**
- 5. Use a model to provide a conceptual framework and scope if it helps (actually experience shows that two are better). Know how to use it, and who's in charge. Don't let model compliance become the primary objective.**
- 6. Don't manage SPI as a project.**
- 7. Measure progress by results, not schedule.**
- 8. SPI is exploratory; some, many even, improvements will not work as you expect. Failures should be regarded as learning opportunities. They are more than compensated for by those improvements that work well.**
- 9. Tactics determine strategy. That is, strategies are valueless until you know what you can actually change in practice.**
- 10. SPI must pay for itself. Demonstrate this or stop.**

OXFORD
SOFTWARE ENGINEERING
LIMITED

9 Spinners Court, 53 West End,
Witney,
Oxfordshire
OX28 1NH

www.osel.co.uk
info@osel.co.uk
Tel. +44 (0) 1993 700878