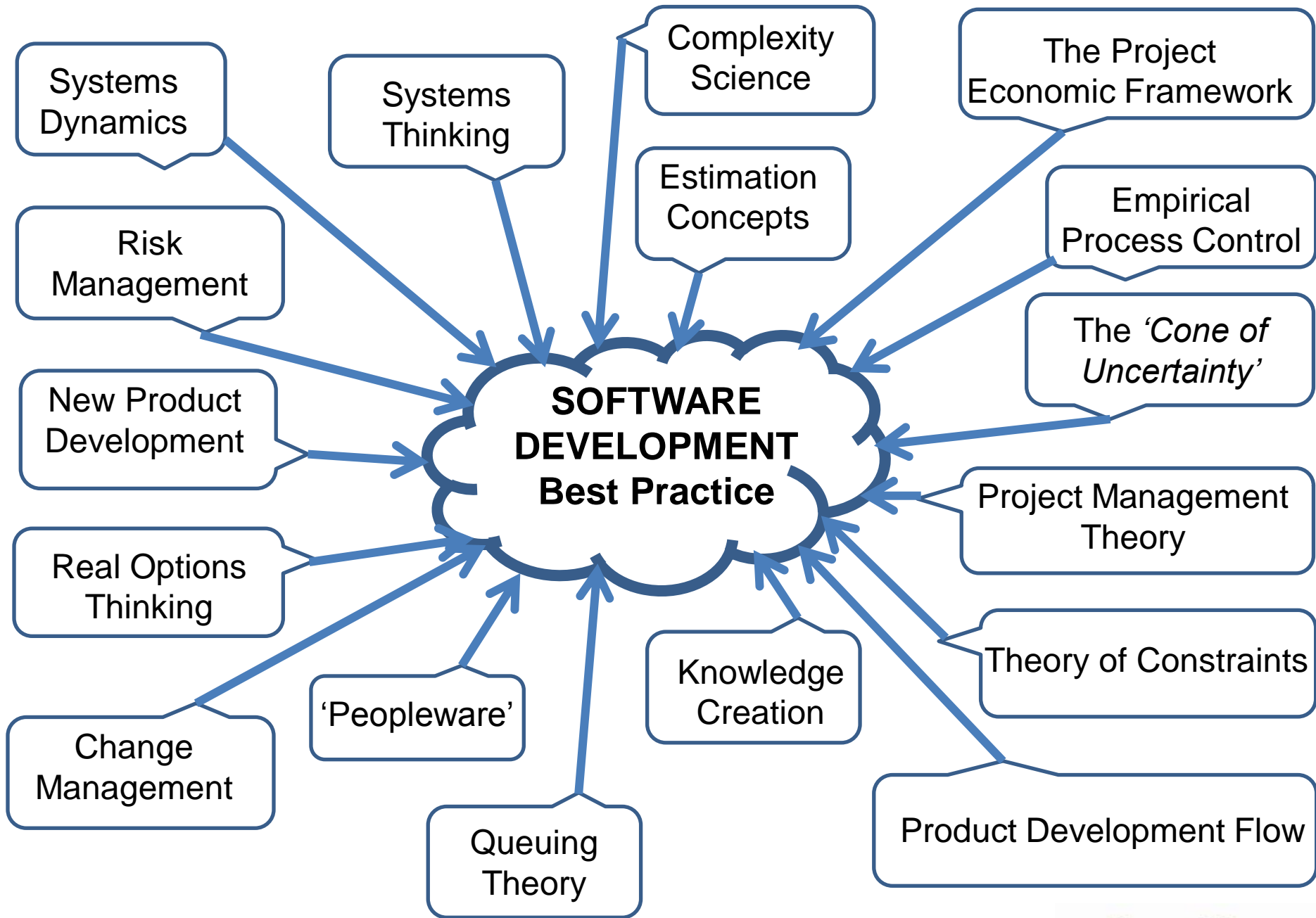


HOW TO CONTRACT FOR AGILE



Intellect
Agile Software Development in Outsourcing and Offshoring
21 March 2012

Susan Atkinson
Legal Director
gallenalliance Solicitors



THE TRADITIONAL CONTRACT STRUCTURE – INHIBITORS FOR AGILE

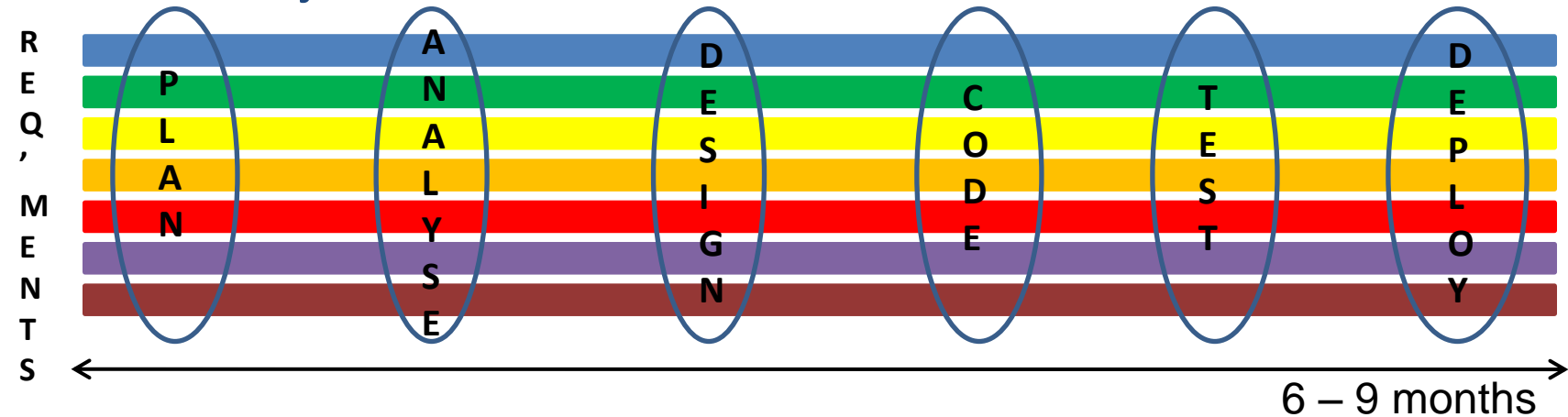
- The substance of the contract:

- the supplier will deliver software which successfully passes the tests to meet the specified requirements by the specified date
- the supplier will analyse THEN design THEN code THEN test ALL of the requirements

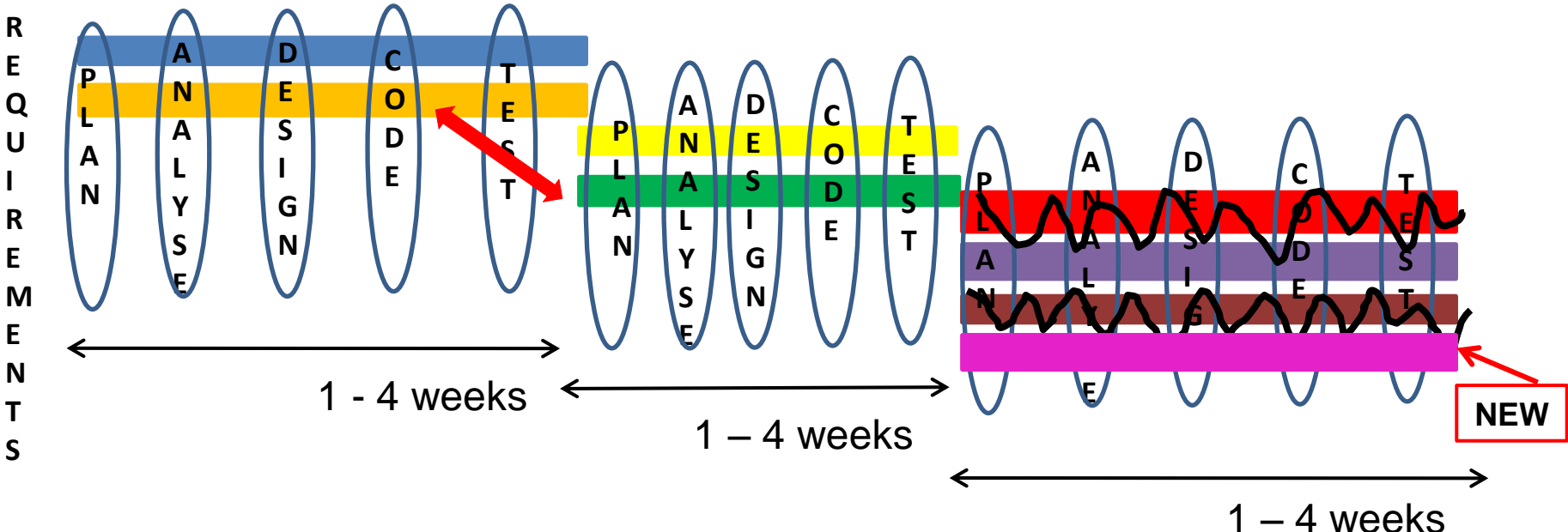
- Fees and payment arrangements
- Resources and personnel (*e.g. key personnel, staff transfer, premises*)
- Remedies (*e.g. limit of liability, warranties, indemnities, step in rights*)
- Intellectual property, data protection and confidentiality
- Contingency arrangements (*e.g. insurance, Force Majeure*)
- Term and termination
- Miscellaneous provisions (*e.g. subcontracting, dispute resolution, governing law*)

COMPARISON OF PROJECT LIFECYCLES

Waterfall Project

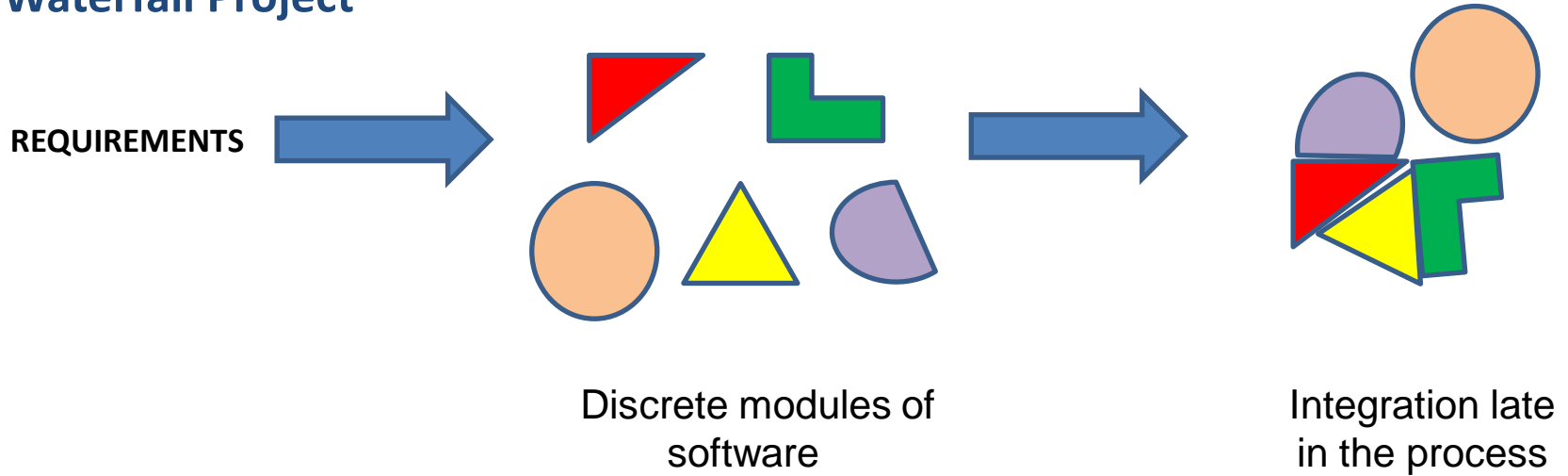


Agile Project

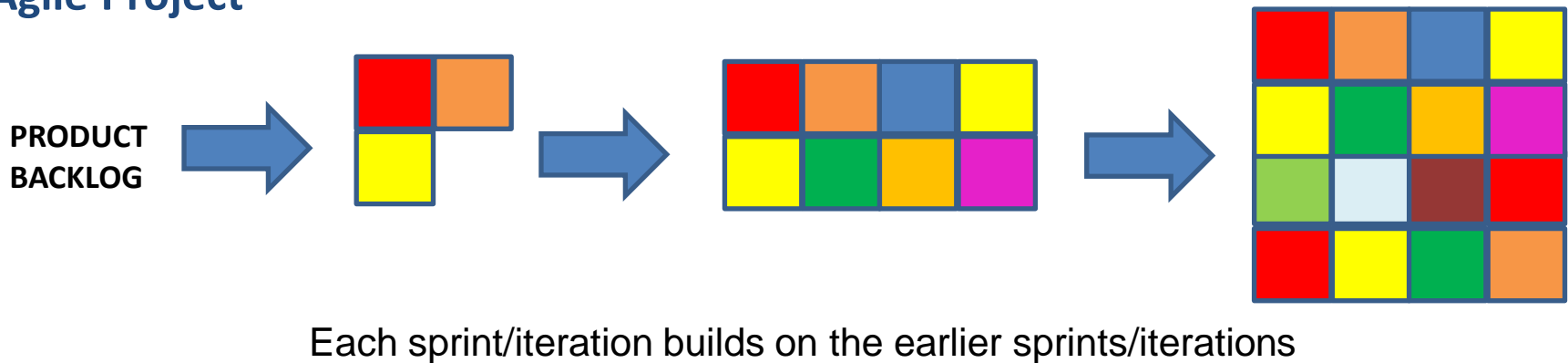


COMPARISON OF SOFTWARE DEVELOPMENT

Waterfall Project



Agile Project



THE TRADITIONAL CONTRACT MODEL: INHIBITORS FOR AGILE DEVELOPMENT

Traditional Contract Model	Agile Approach
The requirements are contractual and specified upfront.	Items of work relevant to the solution are pulled off the product backlog. Only as many work items as can be completed within the budget and time frame are started.
Changes are 'controlled' by means of the change control mechanism.	Change is accommodated by making changes to the product backlog before a work item is started.
Analysis, design, development and testing occur sequentially.	Each work item is taken to completion, with design and development occurring concurrently.
An all-or-nothing solution.	The solution is broken down into work items, which are aggregated into Minimum Viable Products (MVPs).
Constituent 'modules' of software are worked on independently until integration takes place.	There is a continuous working and stable software system.
Testing is used as a contractual tool at the end of the development process.	Testing occurs throughout the development process.
Success is measured by reference to conformance with the plans.	Success is measured by reference to the extent to which the completed work items realise the desired business outcomes of the customer and therefore deliver value.

THE CHALLENGES OF CONTRACTING FOR AGILE

- The traditional contract model is ill-suited for Agile
 - *De Beers v Atos*
- Lawyers cite failed 'Agile' projects as evidence that Agile doesn't work:
 - *BSkyB v EDS* – N.B. RAD and not Agile was used
- Lack of standards:
 - Agile is a philosophy that embraces many different methodologies
 - Many suppliers are jumping on the 'Agile' bandwagon
 - Many organisations don't fully understand why Agile works

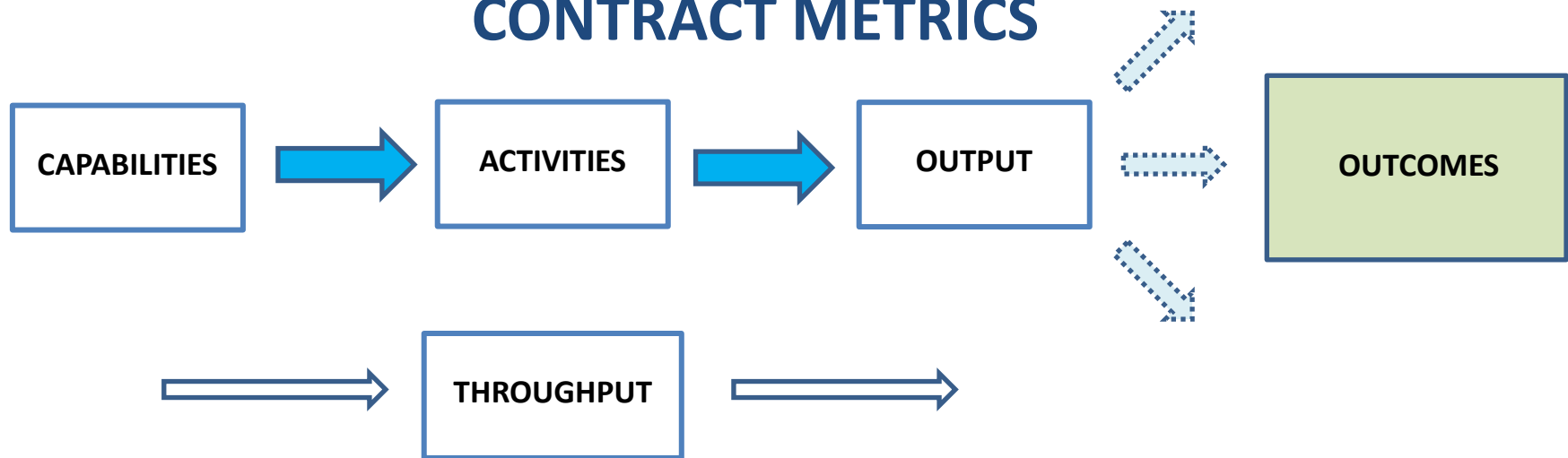
CONTRACT METRICS

*"Perhaps what you measure is what you get.
More likely, what you measure is all you get.
What you don't (or can't) measure is lost."*

H Thomas Johnson, 2006.

- Examples of measuring the wrong thing:
 - ***Train journey times*** – measurement of the performance of individual trains advertised as passenger services against their planned timetable \Rightarrow scheduled journey time increased.
 - ***'Houston Miracle of the Texas School System'*** – performance metrics to hold educators accountable for results \Rightarrow schools dropped low-achieving students from the rolls.
 - ***Call centres*** - measured on the number of ringtones before picking up the phone \Rightarrow customers put onto hold.

CONTRACT METRICS



‘Capabilities’ – technical, financial or human resources that the customer and the supplier invest in the development process.

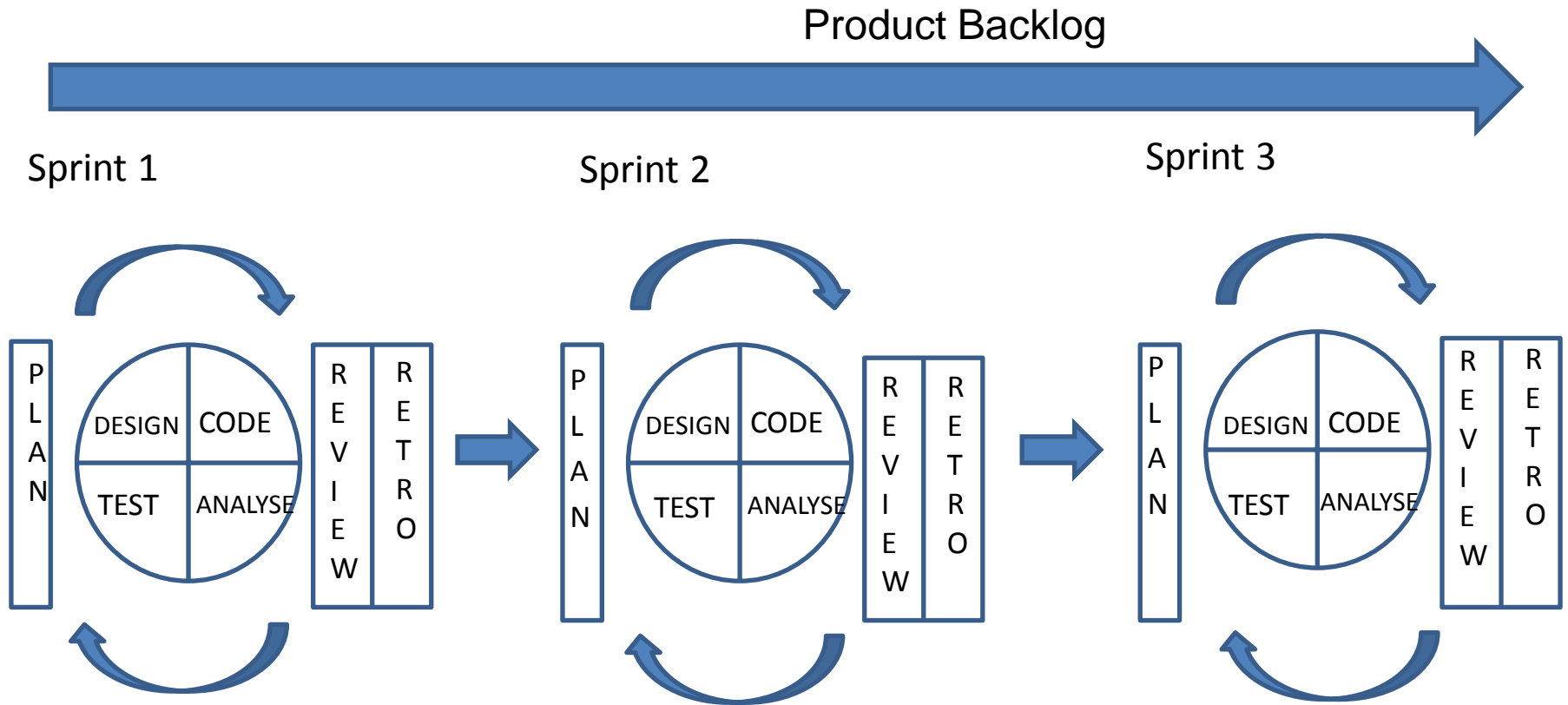
‘Activities’ – activities that the supplier undertakes to transform the capabilities into the output.

‘Output’ – the direct result of the project which is intended to contribute to and facilitate the achievement of the outcomes.

‘Outcomes’ – the high level objectives or goals that the customer wishes to achieve and which will add value to the customer.

‘Throughput’ – measure of the total number of units of work completed over a set period of time (Agile) or the amount of time taken to complete a unit of work i.e. the cycle time (Lean).

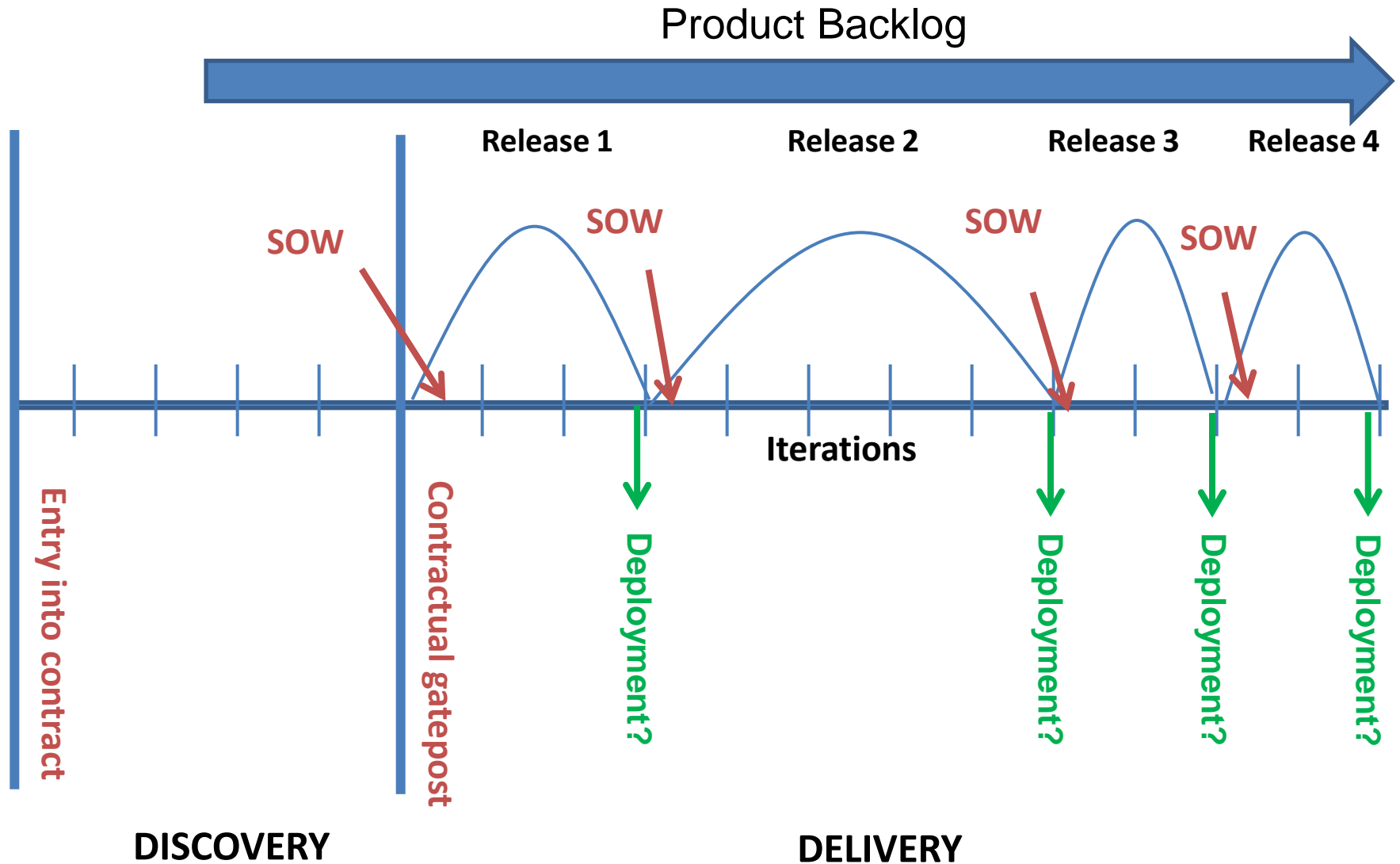
ACTIVITIES – SCRUM MODEL



ACTIVITIES – PROS & CONS

- ***‘Activities’*** - activities that the supplier undertakes to transform the capabilities into the output.
- Pros:
 - More accurately reflects the ‘Agile’ process than the traditional contract model does.
- Cons (main ones only):
 - Inconsistent with the principles of Agile which are ‘results-focused’.
 - Completion of the activities will not necessarily result in high quality output or achieve the customer's target outcomes.
 - The measurement of activities incentivises the creation of more activities.

OUTPUT & THROUGHPUT - MODULAR DEVELOPMENT MODEL



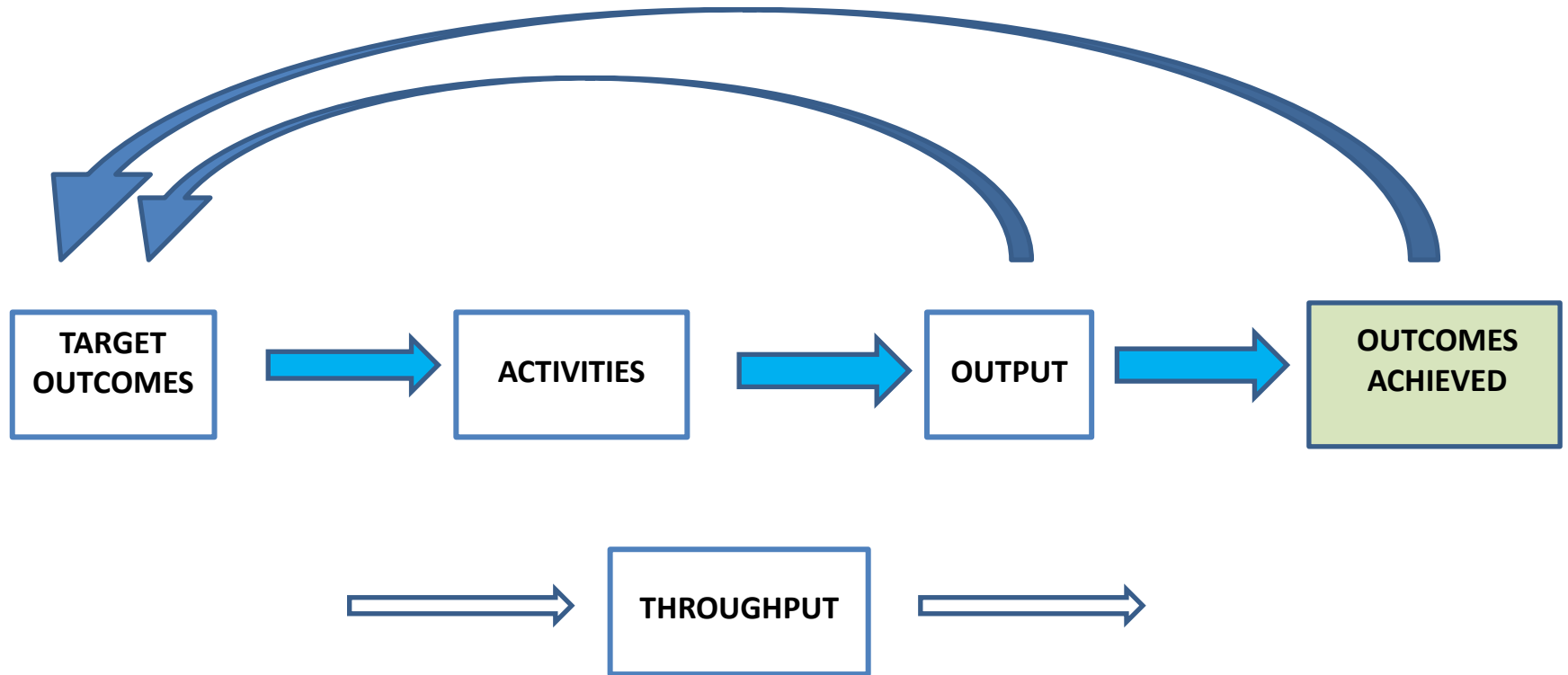
OUTPUT – PROS & CONS

- **‘Output’** - the direct result of the project e.g. products, services, deliverables.
- Pros:
 - Familiarity.
 - Appears to facilitate the management of budgets and schedules.
 - Mirrors current procurement models.
- Cons (main ones only):
 - Delivery of the output will not necessarily achieve the outcomes.
 - The measurement of output incentivises the creation of more output – this potentially increases costs and the time to market.
 - The customer inevitably ends up micro-managing the development process – labour-intensive and requires development expertise.

THROUGHPUT – PROS & CONS

- ***‘Throughput’*** – measure of:
 - number of units of work completed over a set period of time e.g. function points or story points (Agile); OR
 - amount of time to complete a unit of work e.g. cycle time (Lean).
- **Pros:**
 - Increased flexibility.
 - Easy to measure.
- **Cons:**
 - Delivery of throughput will not necessarily result in high quality output or achieve the customer's target outcomes.
 - The measurement of throughput incentivises the creation of more throughput and may lead to a compromise in the quality of the output.
 - The customer inevitably ends up micro-managing the development process – labour-intensive and requires development expertise.

OUTCOME-BASED DEVELOPMENT



OUTCOMES – PROS & CONS

- **‘Outcomes’** - the high level objectives or goals that the customer wishes to achieve and which will add value to the customer.
- Pros:
 - A structured approach for focusing on the customer's strategic plan.
 - It is possible to incentivise the supplier to take a more direct approach to achieving the target outcomes of the customer.
 - Outcomes are better suited to withstand change.
- Cons:
 - Lack of familiarity.
 - Outcomes are not as straightforward as other contract metrics – outcomes must be clear and quantifiable.
 - Transitioning to Agile at the same time as moving to outcome-based development may be too big a cultural change for some organisations.

CONCLUSION

- Contracting for Agile is hard.
- Not only is Agile an evolutionary process, but the contract model is also evolutionary. As organisations become more experienced at using Agile, the type of contract they wish to use changes.
- We are going to see a paradigm shift in the type of contract model used for Agile.
- In the short-term we are probably going to see a number of Agile project failures in the commercial context. There is a lack of standards for Agile, and this is open to abuse.

THANK YOU



Susan Atkinson
gallenalliance Solicitors

9 Devonshire Square

London EC2M 4YF

United Kingdom

Tel: +44 20 7084 6392

Email: satkinson@gallenalliance.com