Contracting for Agile Projects.

Amy Ryburn
Damien Steel-Baker

September 2022

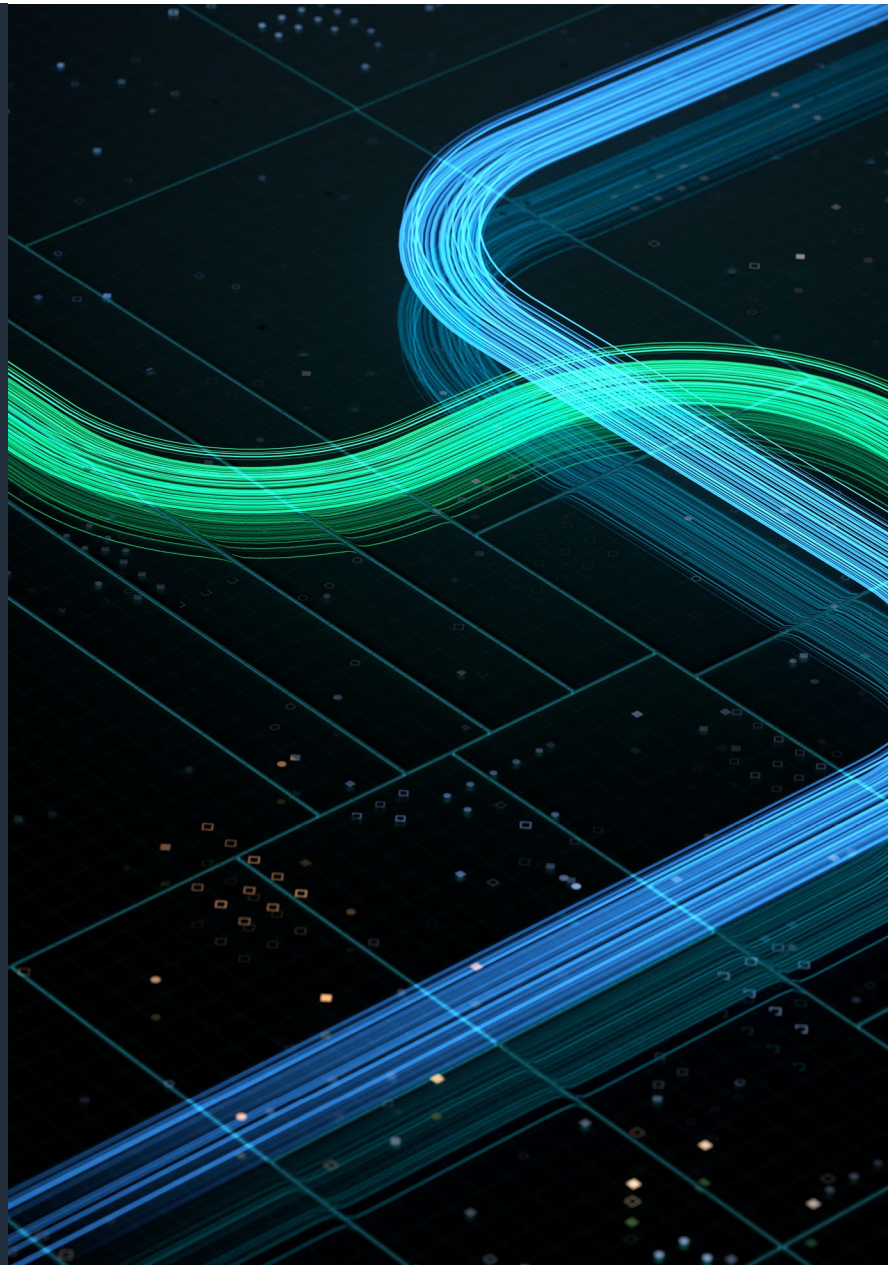
“Agile software development methods are now being widely used in the IT sector and are increasingly being advocated as preferable to the traditional waterfall development model.”

Bird & Bird,
“Contracting for Agile software development projects”,
Position paper 2012, page 1.

A standard traditional Master ICT services agreement will not be appropriate for most agile projects without revision.

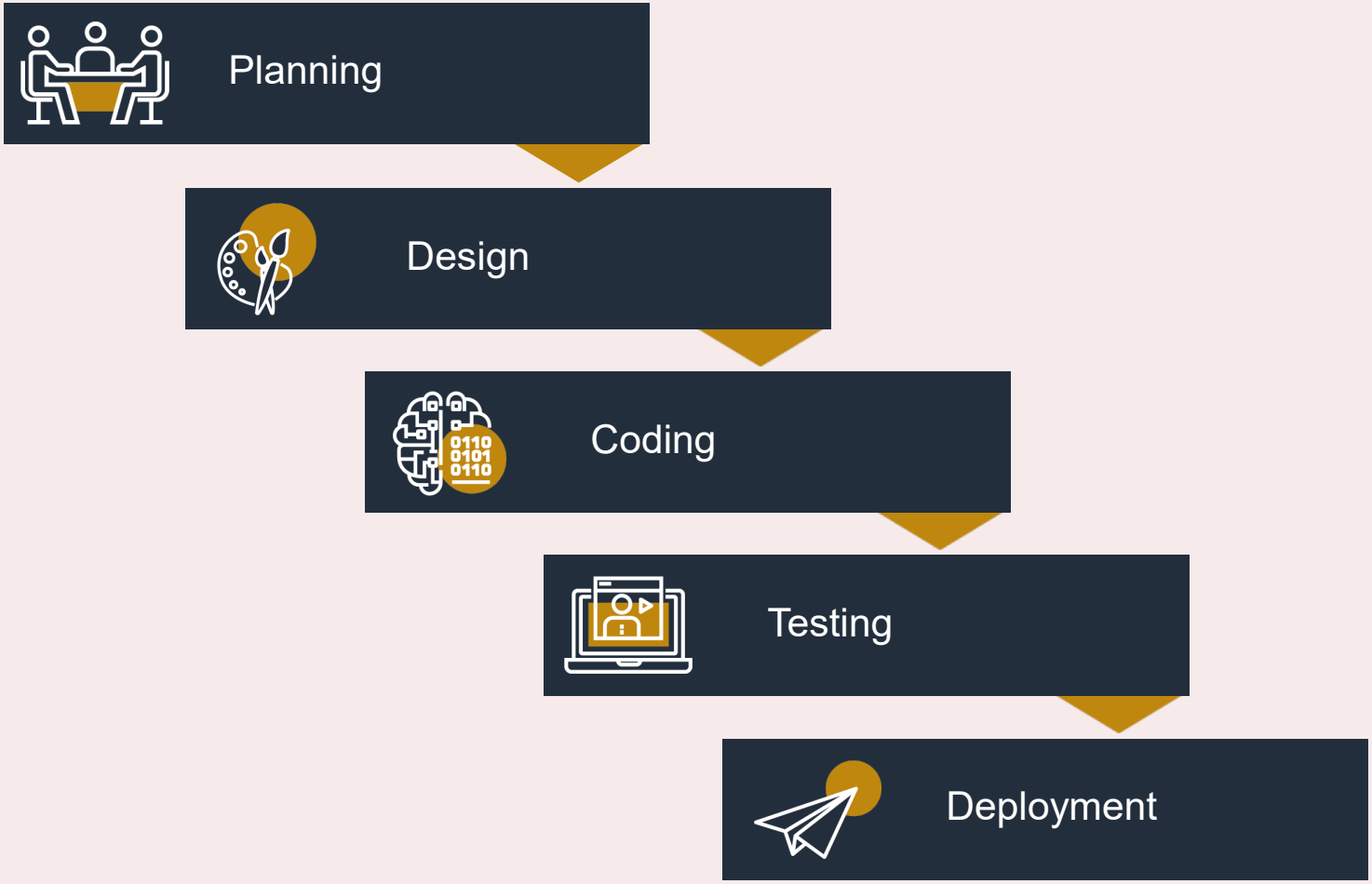
Typical approaches to contracting for agile projects

- The time and materials engagement without commitment to outcomes
- A standard MSA with Statements of Work referencing the use of “agile methodology” and potentially including some agile terms



So what is an appropriate contract for an agile project?

The waterfall model



Waterfall project methodology - are the benefits just an illusion?

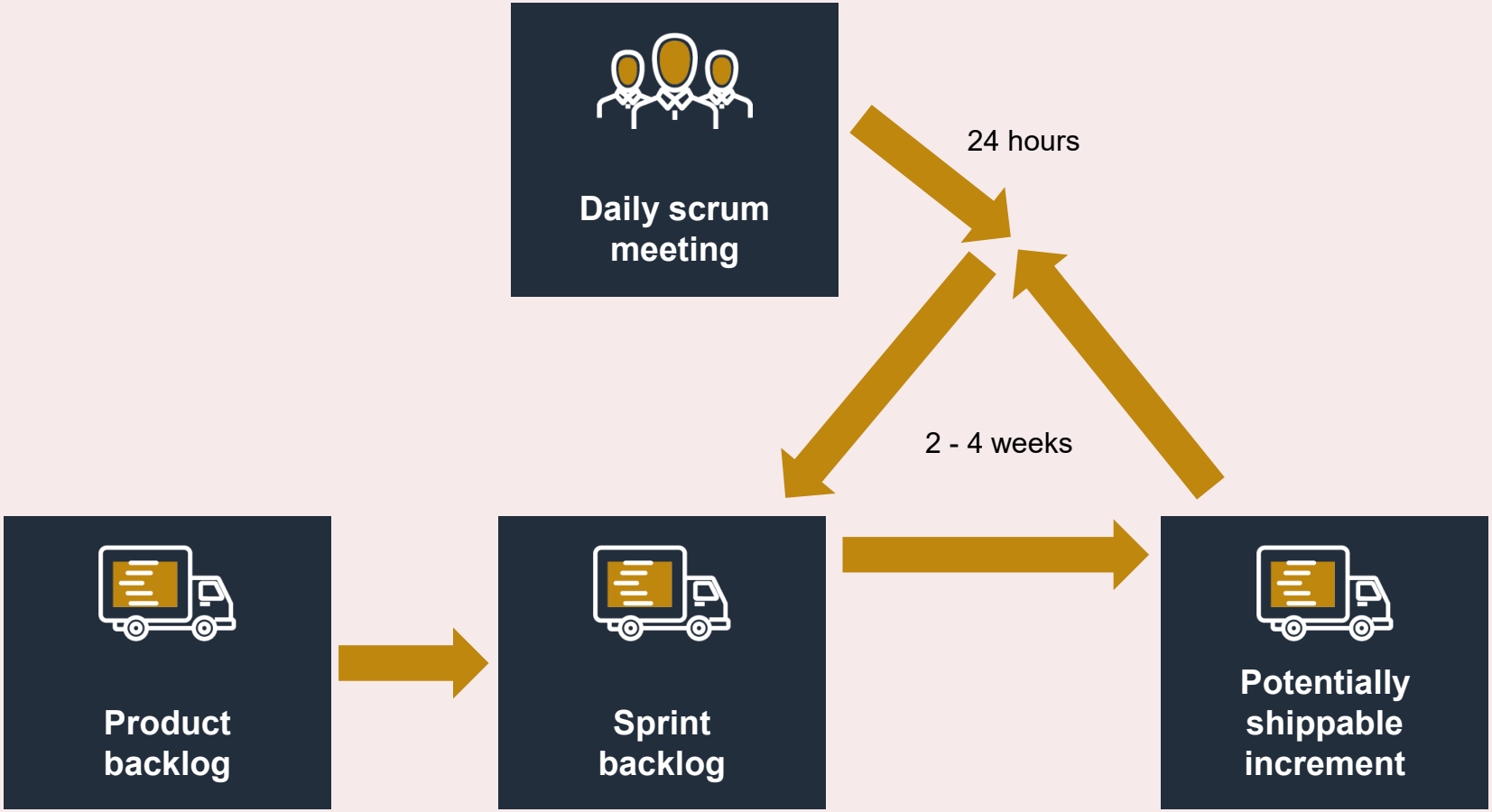
BUDDLE FINDLAY

Agile values

Manifesto for Agile Software Development - 2001

Individuals and interactions	over	processes and tools
Working software	over	comprehensive documentation
Customer collaboration	over	contract negotiation
Responding to change	over	following a plan

The scrum model



Common terms and what they mean

- Product owner
- Scrum master
- Development team
- Retrospective sprint review
- User stories/epics
- Product backlog/sprint backlog
- Potentially shippable increment
- Minimum viable product/Minimum marketable release
- Burn down/up chart

Key advantages

- No upfront detailed requirements, work out requirements as you go
- Use of non-technical "user stories"/focus on objectives rather than technology
- Allows customer the ability to prioritise (and later re-prioritise) what is important to the customer
- Limit costly change control
- Speed - an early usable product or a fast failure
- Greater customer involvement/business buy in
- There is some evidence to suggest that agile projects fail less frequently and are more likely to deliver something the customer actually wants

Key challenges

Certainty of scope

As requirements are not fixed at the outset there is reduced certainty of scope.

Certainty of time

Failed features are put back into the backlog (rather than seen as breaches). There is a risk that the project can continue for an indefinite number of sprints.

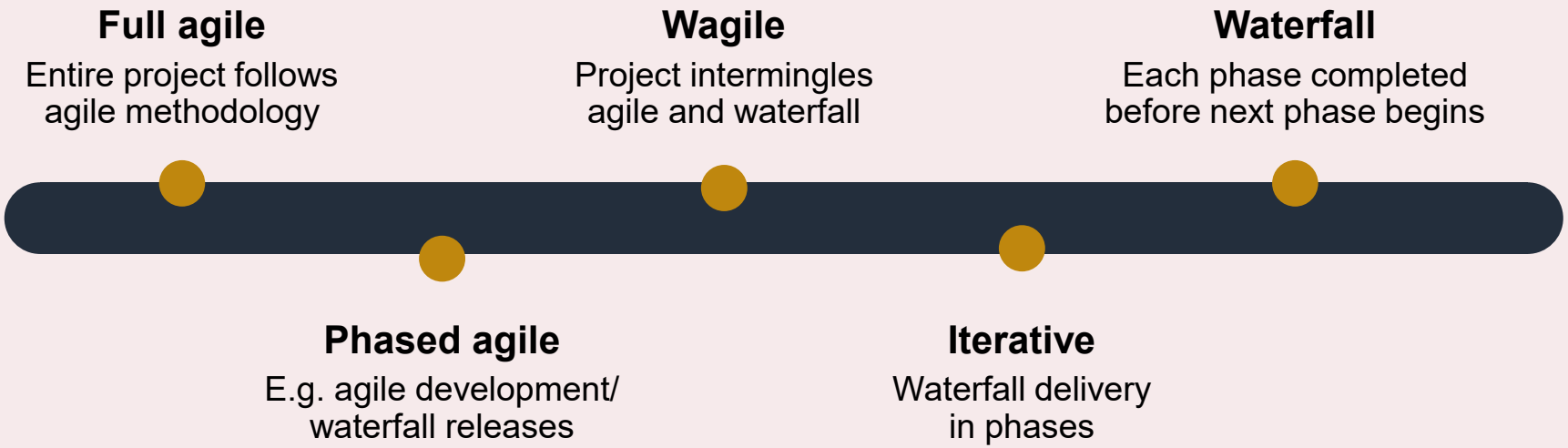
Certainty of cost

Agile projects are often done on a time and materials basis. Lack of fixed requirements at the outset may make it very difficult to give a fixed price for some or all of the work.

Certainty of risk allocation

Joint project teams can mean that when something goes wrong it is difficult to allocate responsibility/hold the other party to account.

A continuum


Full agile
Entire project follows agile methodology
Wagile
Project intermingles agile and waterfall
Waterfall
Each phase completed before next phase begins
Phased agile
E.g. agile development/ waterfall releases
Iterative
Waterfall delivery in phases

Contractual options on the continuum

- Time and materials - consultancy contract
- Time and materials - process and roles agreed
- Time and materials - process and roles agreed AND measures of success linked to remedies (payments potentially linked to milestones)
- Wagile - plan based but using elements of agile (e.g. daily stand-ups involving both parties, sprints)
- Iterative waterfall
- Some agile, some waterfall
- Ultimately an “agile” contract will tend to be a more process-oriented document because the substance of the project is continually refined. It is critical that the process is well defined

Key questions to ask customer clients

- Why agile? What are you hoping to get out of using the process?
- Do you have a fixed set of requirements? Are these likely to change?
- Do you favour certainty or innovation?
- Do you have a fixed timeframe or budget or some room to move?
- Could you/would you walk away at any point?
- Have you worked with the supplier before?
- Are you resourced well internally to support an agile process? Has it been used internally already?

Key pitfalls and potential mitigants

- **Poor quality**: Focus on acceptance criteria for each sprint/iteration. Limit the number of times a feature can go back into the backlog. Require the supplier to pay for re-working failed requirements after [X] attempts. Warranty that each iteration (or sprint) outputs will work together with previous iterations. Note that the supplier may want some control over the environment/team
- **Lack of commitment to delivery**: Minimum system delivery by fixed dates. Termination for convenience rights/stage gates (and appropriate governance to decide whether to exercise them) for when projects keep dragging
- **Lack of financial incentives**: Portions of payments linked to achievement of system requirements. Fixed price for minimum requirements. Pain/gain share models
- **Confusion over who does what**: Allocate and record sprint responsibilities. Independent scrum master. Processes regarding record keeping

Key mitigant: The Customer needs to understand/work with the process and stay actively involved throughout the whole project

Supplier frustrations

- Customers who want the benefits of flexibility but still want fixed milestones, fixed price and/or a commitment to a meet a significant set of (vague) requirements
- Customers that treat everything as essential, not allowing any flexibility in what is to be delivered, despite encountering change along the way
- Customers who restrict the agile process (e.g. hindering early and regular delivery through bureaucracy)
- Customers who don’t resource the project from their end
- Customers who want total control but expect the supplier to take total responsibility
- Customers who treat all defects as a supplier failure

Customer frustrations

- Suppliers who over-sell and under-perform and use the process as an excuse (including poor effort/story point estimation which makes it hard for a customer to know how and when to prioritise)
- Suppliers who push all risk to the customer - reaping all the benefits themselves
- Suppliers who ignore the process or say they work to a particular process but then don’t follow it
- Suppliers who treat all defects as just part of process (i.e. never a supplier failure)
- Suppliers who have too much focus on the day-to-day without enough focus on the overall vision/design (i.e. lack of expert guidance/leadership)

Dealing with disputes

Not following the process can cause big problems	Joint project teams makes allocation of blame difficult	Gathering evidence is expensive and time-consuming	Reluctance to call time can impact on your remedies
While significant time may be spent defining the process, often it isn't followed but without that there is very little protection. It becomes very important that the process documented will be followed.	Failure to actually document who was doing what at a detailed level. Thought should be given to how the parties will record who is to do what as they progress.	Evidence that will help your lawyers determine your chance of successfully pursuing remedies isn't in the contract and perhaps not in writing at all. Consider reporting and governance obligations carefully to ensure there will be some written evidence of failures.	Parties keep going when project goes off the rails but don't use their remedies. You may have good remedies (e.g. termination rights) but no appetite to use them.

Conclusion

- Don’t use a standard template without at least considering changes
- Ask lots of questions and don’t be intimidated by the jargon
- Ensure the contract reflects processes the parties will actually follow
- Set the project up for success and to identify failure early (good governance)
- Ensure active engagement from both parties
- Be mindful of the risks

BUDDLE FINDLAY

Auckland • Wellington • Christchurch