copenhagen goto; conference

A CASE STUDY IN LARGE SCALE LEAN-AGILE ADOPTION

Chris Berridge
Maersk Line

INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

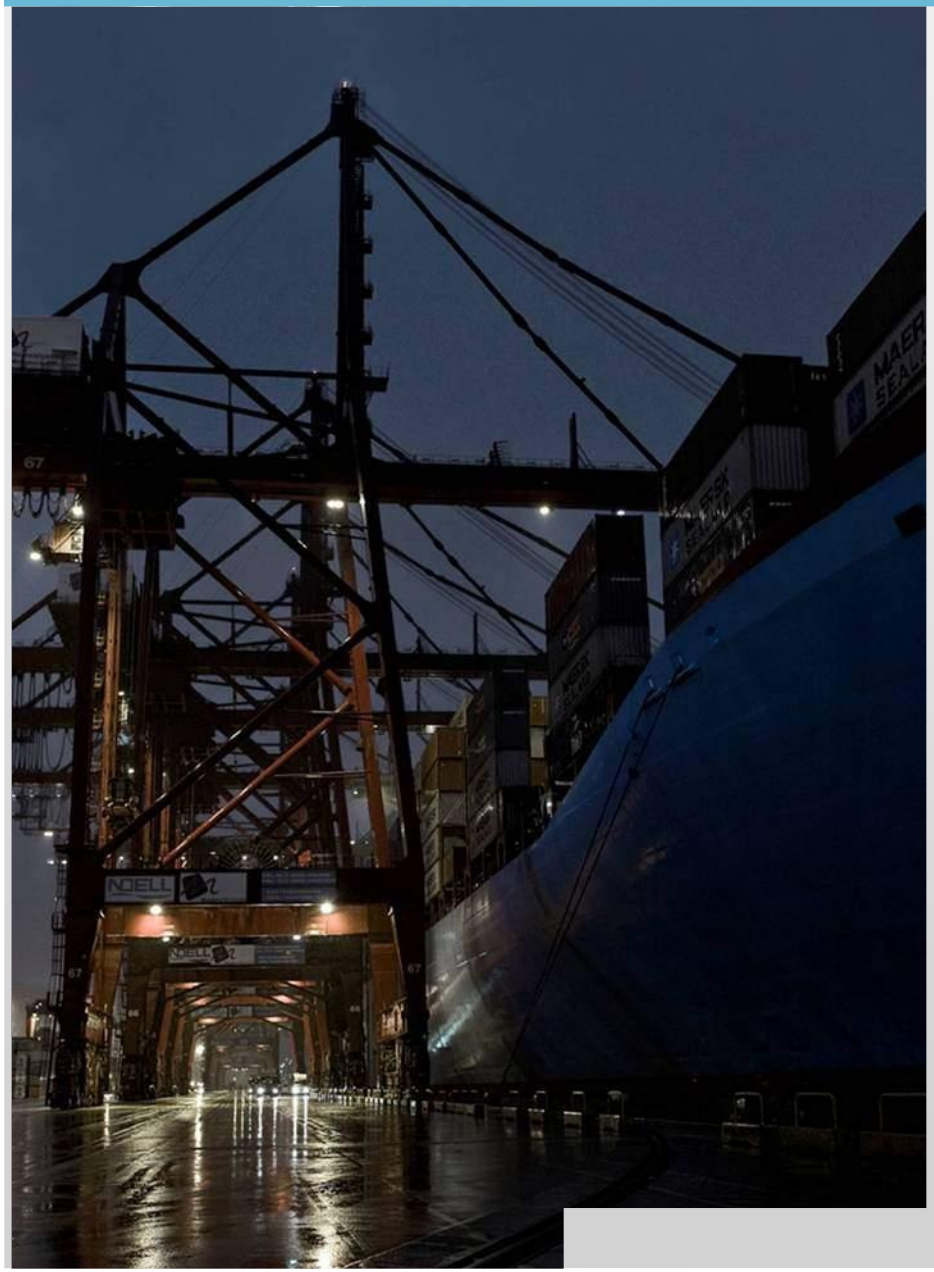
gotocon.com

About Maersk Line

- Worlds largest container fleet
- Truely global business
  - 325 offices in 125 countries
  - 25.000 employees (7,600 seafarers)
- 14.5% world market share [1]
  - 570 container vesssels
  - Turnover $26 billion [2]

[1] Source: Alphaliner Jan 2011
[2] Source: Annual Report 2011

MAERSK LINE



Fragmented IT Landscape

- Thin outsourcing model
- Tier 1 vendors only
- 2,500 applications
- Core applications are tightly coupled
- 23,000 bookings/day

MAERSK LINE

How we started our lean-agile journey?

New
Project, Platform, Team

Revolutionary

+

Existing
Project, Platform, Team

Evolutionary

X-LEAP
Trust, Cared for & Pleased

Lean Product Development

emergn

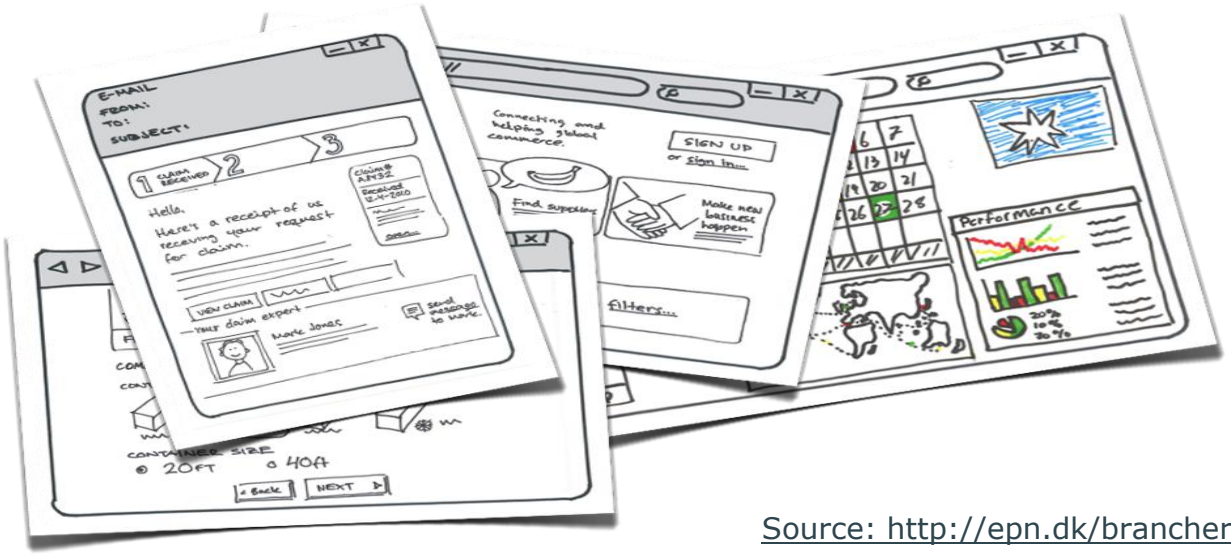
MAERSK LINE

X-leap: The goal

Under Maersk Lines paraplystrategi - streamLINE - er der i værksat en række initiativer, der sikre at rederiet bliver endnu mere konkurrencedygtige gennem industriens bedste leveringssikkerhed, fortsatte CO2-reducerende initiativer og sidste men ikke mindst ved at sætte kunden i fokus

X-Leap er Maersk Lines største og vigtigste af disse programmer.

Formålet er at gøre det ligeså enkelt at booke en container hos os som en bog hos Amazon.com



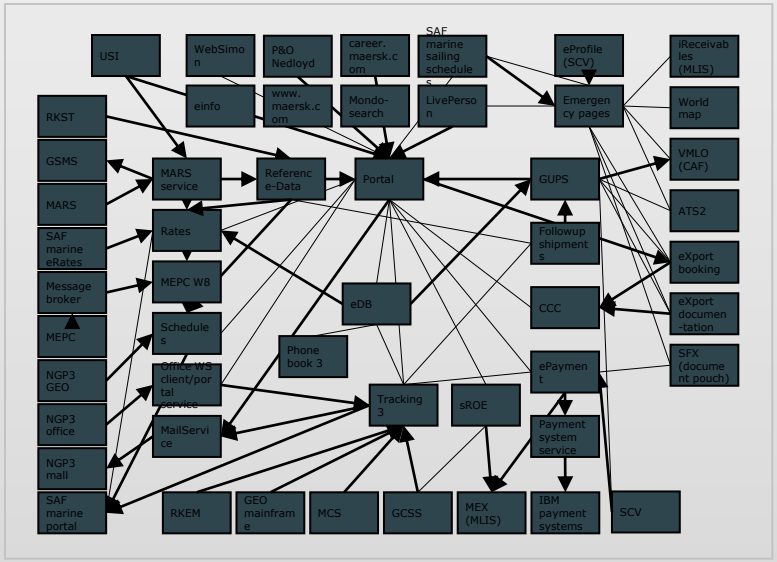
Maersk Line CEO (at the time)

MAERSK LINE

Source: http://epn.dk/brancher/transport/skib/article2069838.ece

X-leap: How we sold agile to our stakeholders

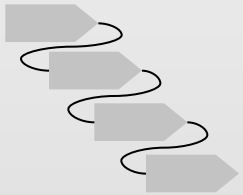
Maersk is complex



- 100's of backend systems
- Convoluted and unstable application architecture
- Inconsistent master data
- High product complexity
  - More than 20 000 lines in some contracts
  - More than 500 commodity types

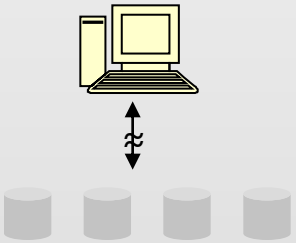
Two delivery approaches are common

1. Waterfall



No customer facing functionality for the first 18-24 months

2. Prototyping

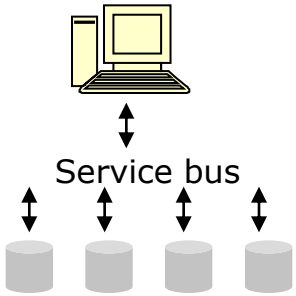


Lots of functionality early, but no connection to backend

Our approach is fundamentally different

Agile SOA

Minimal set of customer facing functionality delivered with true backend connections as early as possibly (in our case 9–10 months)



MAERSK LINE

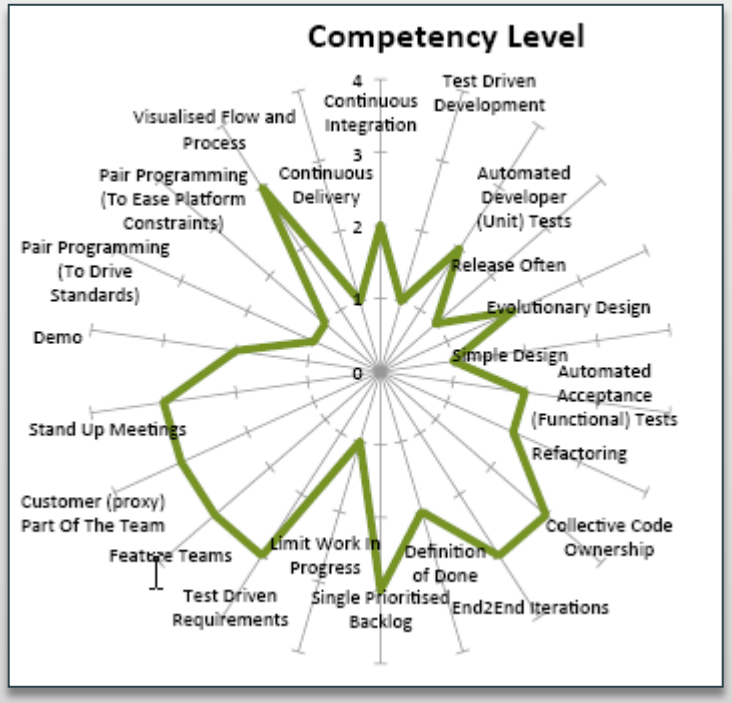
X-leap: What we got right from the outset

- Strong customer focus
  - Clear customer experience vision created
- Co-location
- Shared Key Performance Indicators for whole team
- Onboard experienced people
- Willingness to experiment with new approaches
- Great senior leadership support

MAERSK LINE

X-leap: 22 practices we (now) know that need to master

- Visualised Flow and Process
- Continuous Delivery
- Continuous Integration
- Test Driven Development
- Automated Developer (Unit) Tests
- Release Often
- Evolutionary Design
- Simple Design
- Automated Acceptance (Functional) Tests
- Refactoring
- Collective Code Ownership
- Definition of Done
- End2End Iterations
- Single Prioritised Backlog
- Limit Work-in-Progress
- Test Driven Requirements
- Feature Teams
- Customer (proxy) Part Of The Team
- Stand Up Meetings



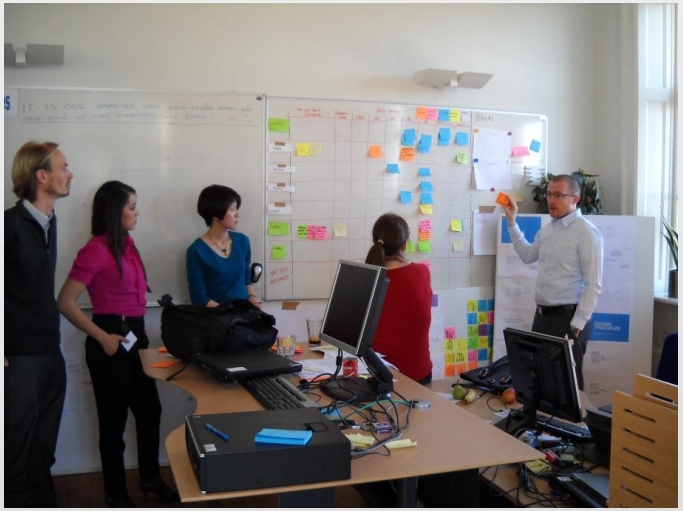
- Demo
- Pair Programming (To Drive Standards)
- Pair Programming (To Ease Platform Constraints)

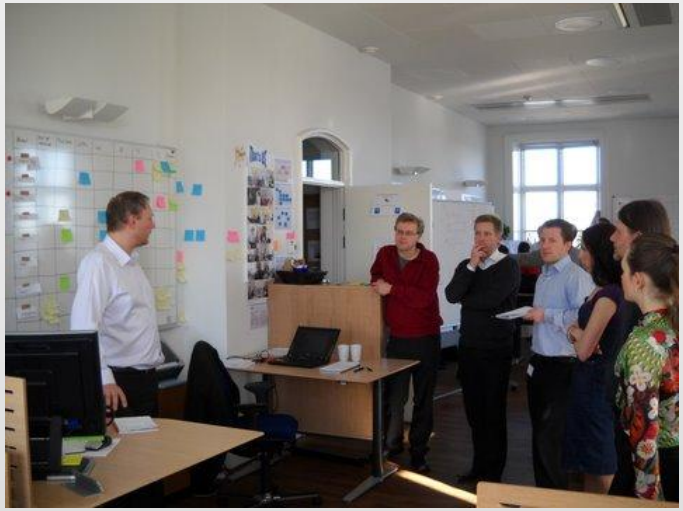
MAERSK LINE

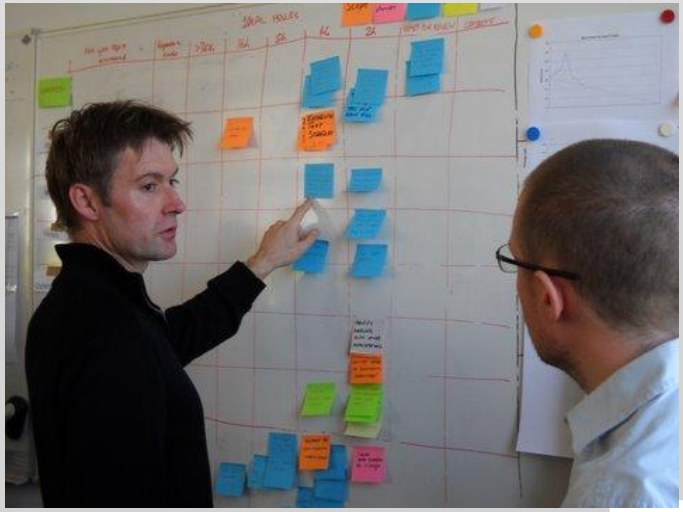
Continuous Delivery Compliance Matrix

Acceptance Criteria	Compliance Level			
	None/ Never	Some/ Sometimes	Most/ Usually	All/ Always
All Code, Configuration, Test scripts, database scema/data migration etc is source controlled.				
All platform config, application config, database schema/data migration is automated.				
Any environment can be built from scratch on demand.				
All environments from development/test stations through to production are built using the same automated process.				
Any single check-in can pushed through all environments with single-click promotion at each stage.				
Binaries are only built once (outside local dev/test builds) before being pushed to subsequent delivery stages.				
All build artifacts can be traced directly back to all source controlled assets that were used to build them.				
There should be an audit log of builds and test runs.				
Source control branching is not used to create long lived branches.				

X-leap: A feature team in action







MAERSK LINE

X-leap: Learnings within team

Manage requirements

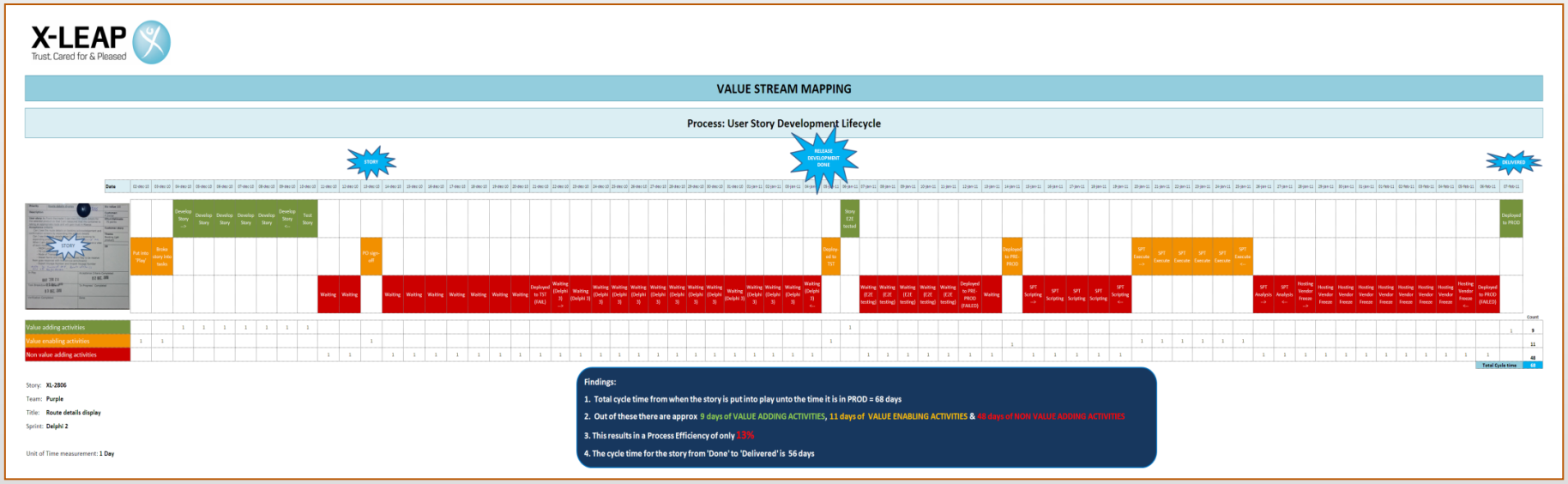
- Prioritise effectively between functional & non-functional requirements
- Break down requirements and agree on what size is appropriate
- Need a process vision to support a customer experience vision

Iteration 0 is surprisingly large

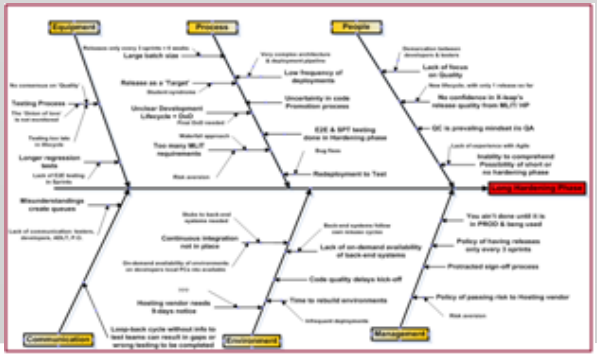
- e.g. Reducing hardening phase took forever

MAERSK LINE

X-leap: Value stream analysis for a feature



X-leap: Root cause analysis for why hardening phase takes so long



MAERSK LINE

X-leap: Learnings within team

Manage the change

- Engage advisors who focus on optimising the whole
- Own and manage practice adoption progress

Minimise thrashing

- E.g. Struggle to measure velocity due to constant changes

MAERSK LINE

X-leap: Learnings outside team

Stakeholders need careful management

- Reluctant to exchange predictability for speed
- Difficult to explain refactoring & technical debt
- High expectations of delivering fast

Dependencies external to the development team are a headache

- Feature teams help but are no silver bullet
- There’s no replacement for good project management to identify and manage external dependencies
- Others have to change their working practice (architects, infrastructure, other applications)

MAERSK LINE

How we are completing the lean-agile journey.

New
Project, Platform, Team

Revolutionary

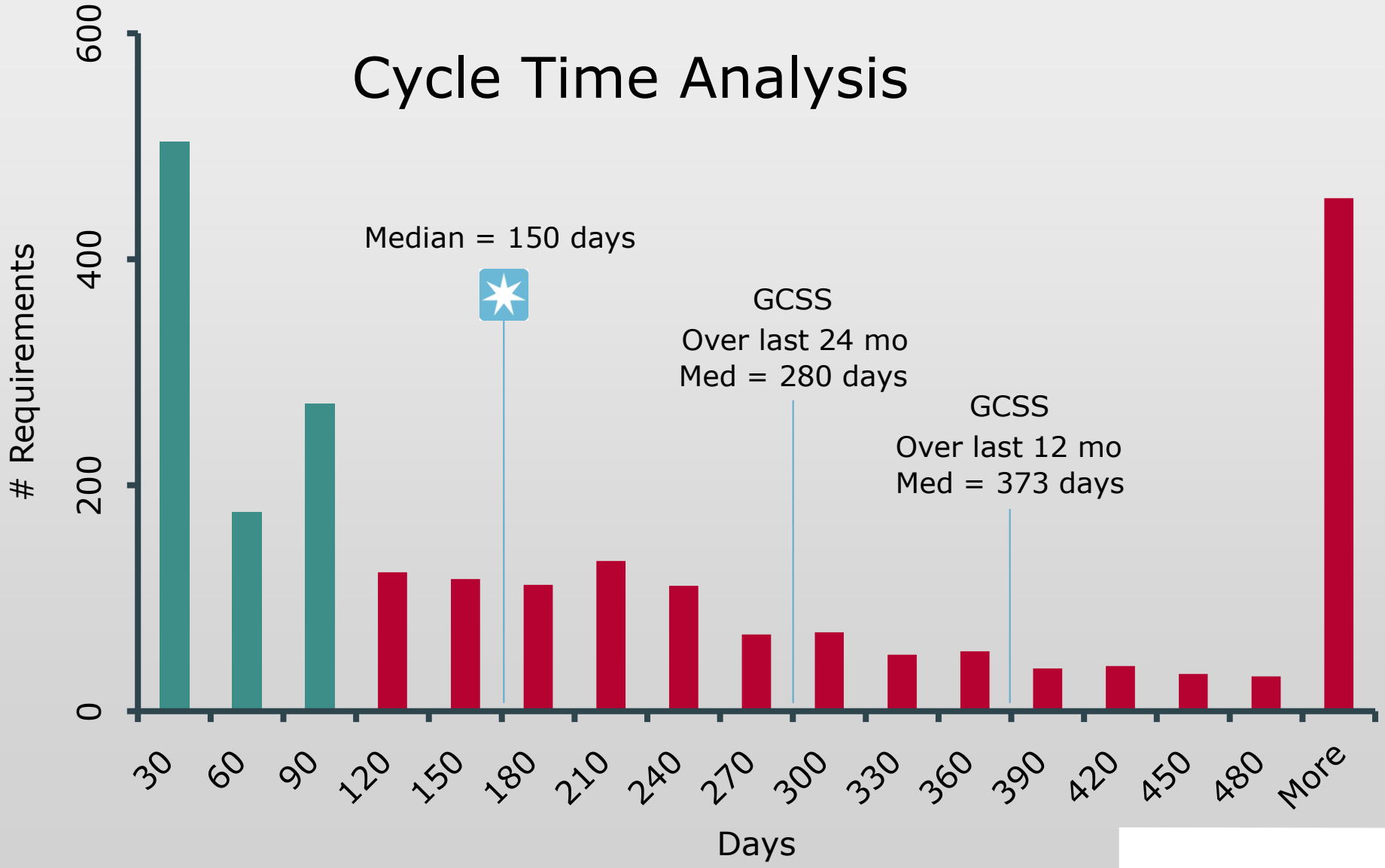
Existing
Project, Platform, Team

Evolutionary

X-LEAP
Trust, Cared for & Pleased

MAERSK LINE

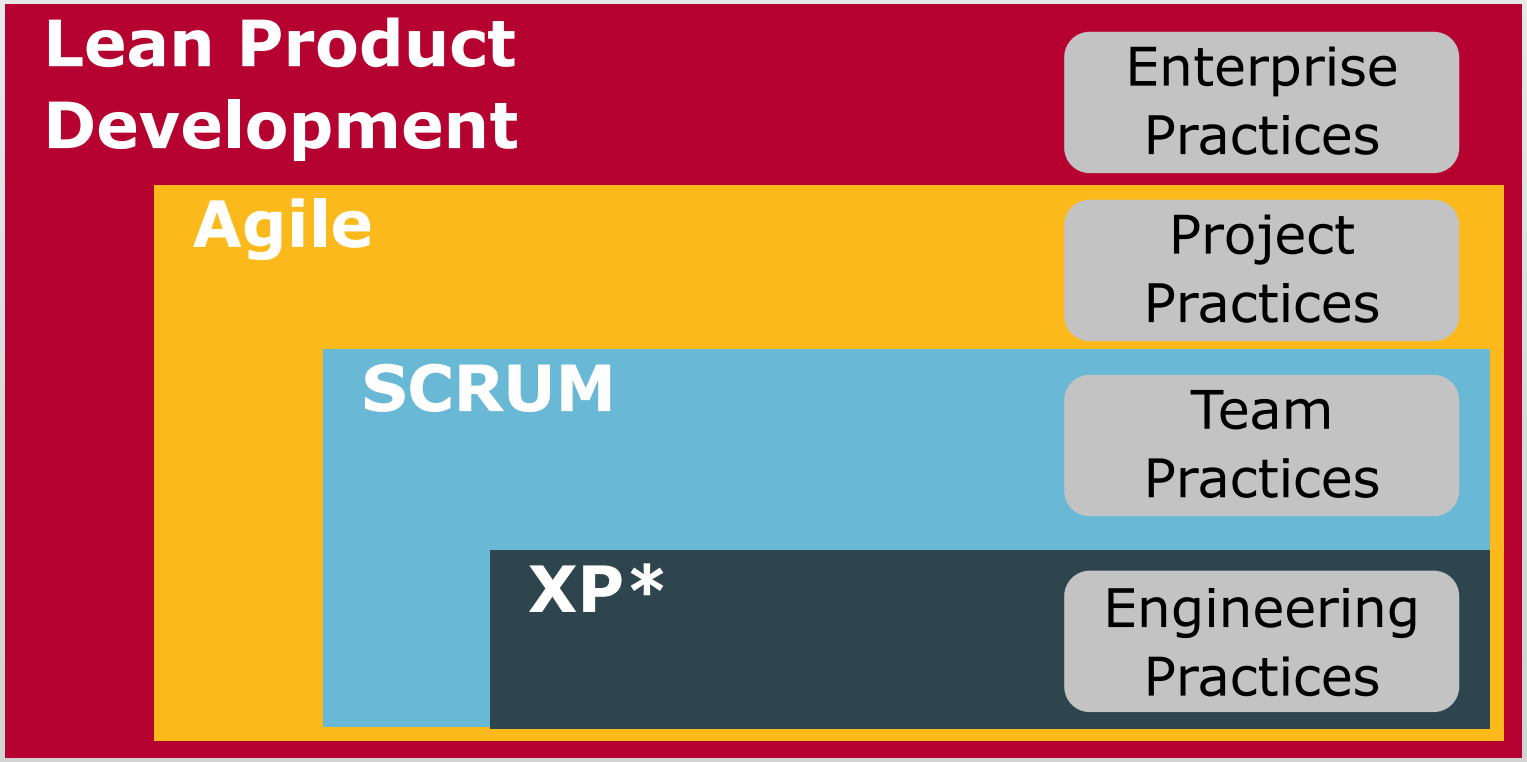
Cycle Time Analysis



Source: Focal Point – requirements that have been put into production over the last 2yrs, measured from date of creation to when set to working-in-production

MAERSK LINE

Framing the methodologies



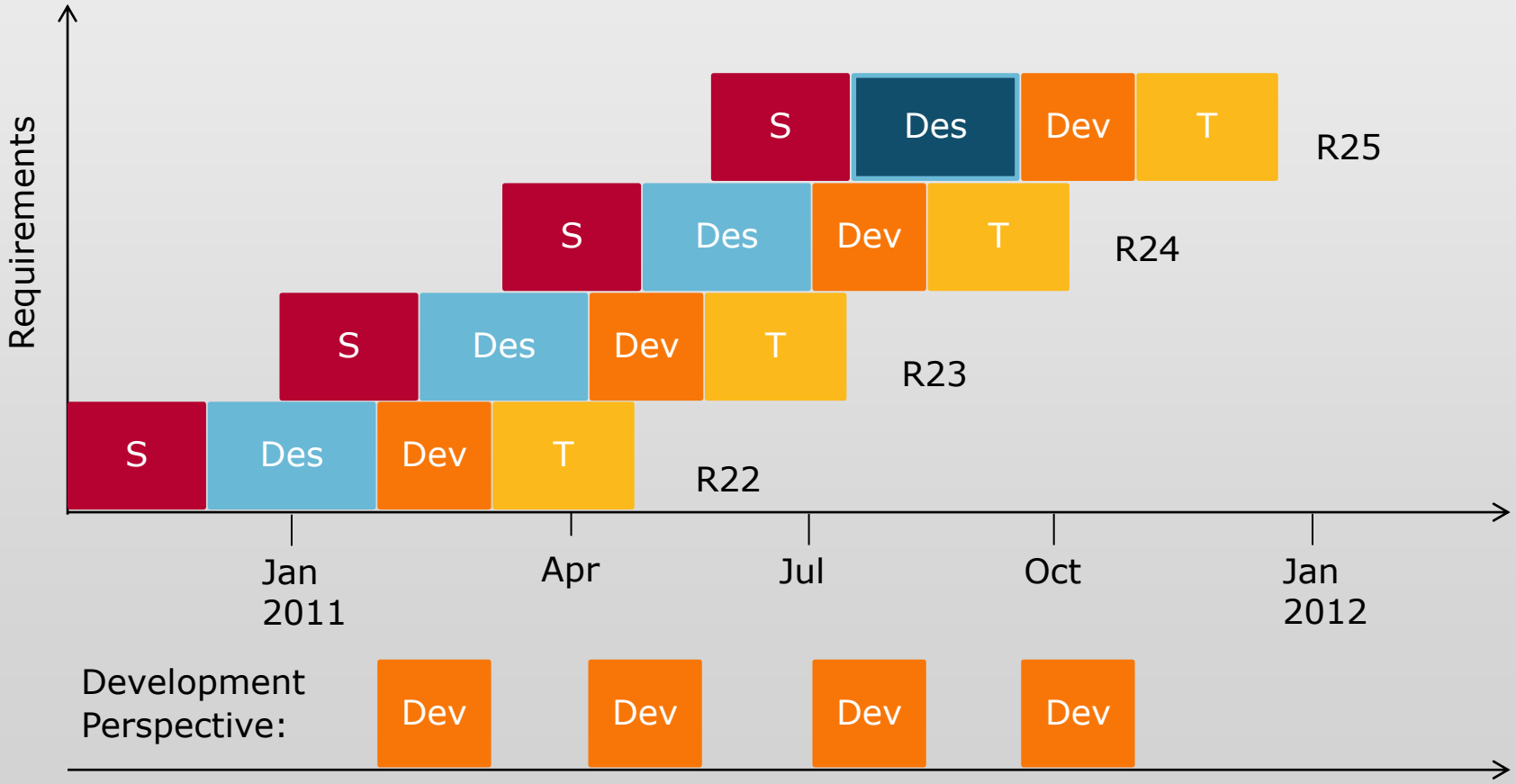
* Extreme Programming

The Starter Pack: 8 selected practices

MAERSK LINE

GCSS: Release Frequency

The effect of creating large release batches upstream

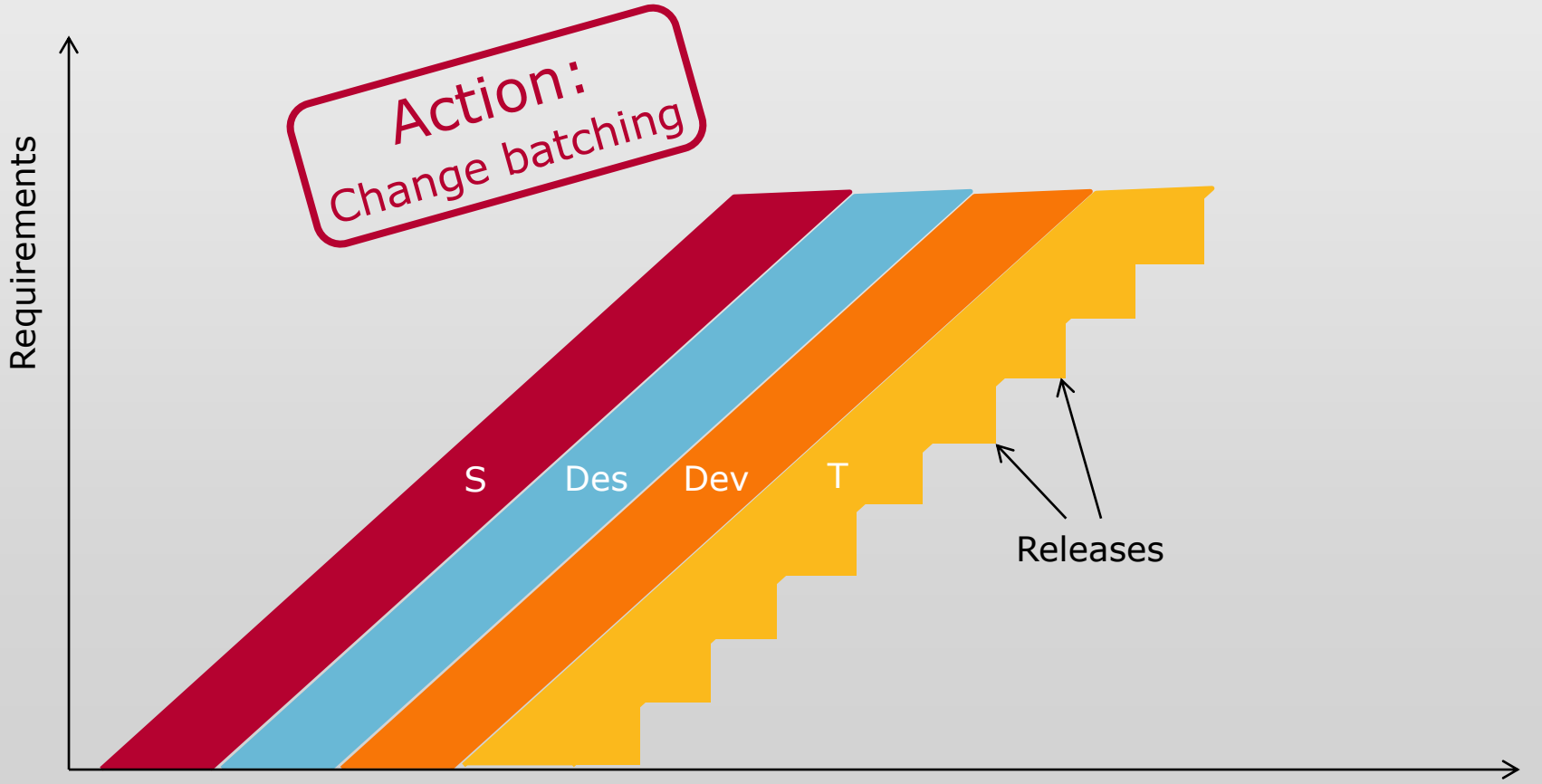


Estimated ~10,000 hours of idle time in 2010

MAERSK LINE

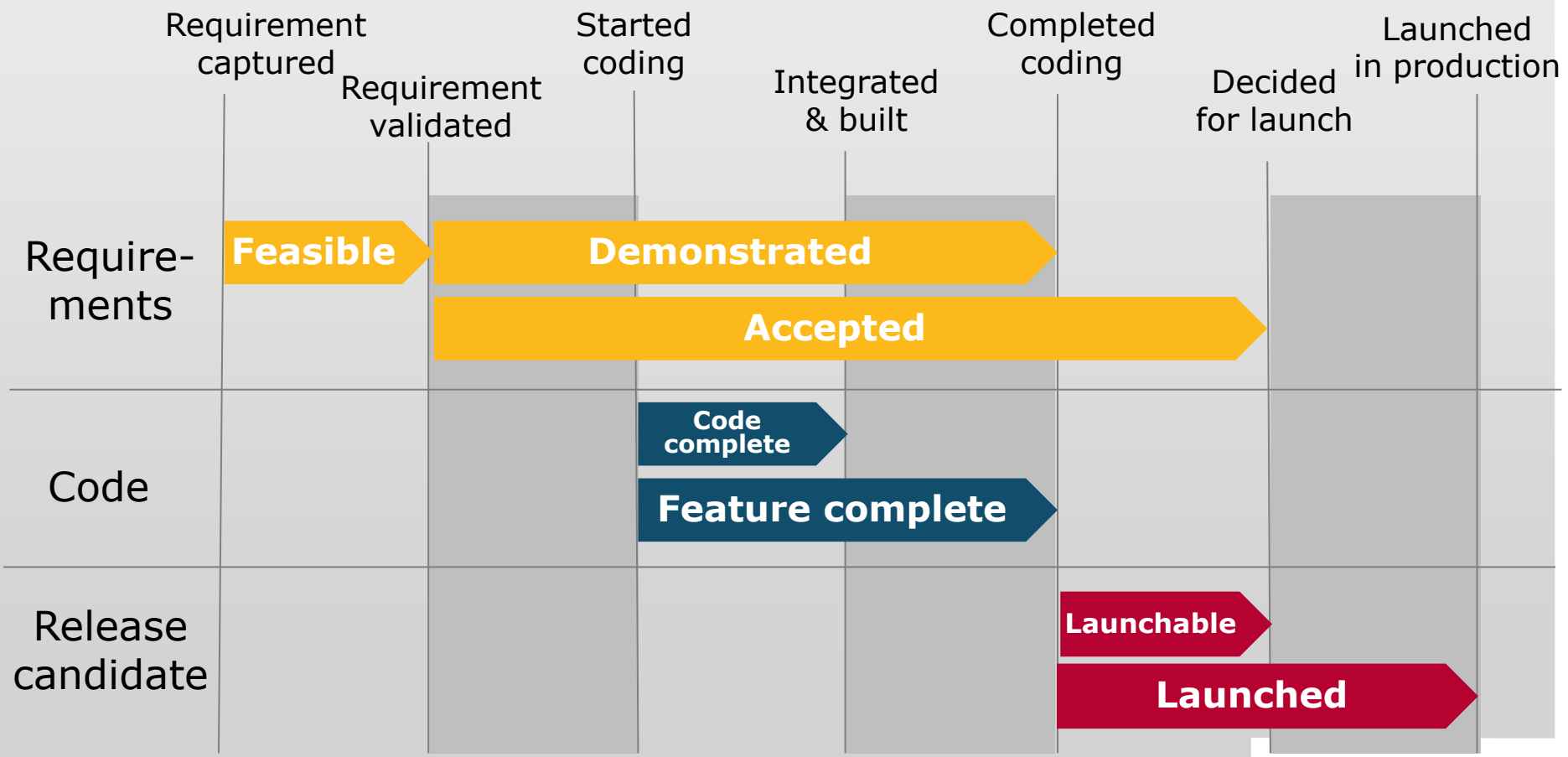
GCSS: More Frequent Releases

Enable the smooth flow of requirements



MAERSK LINE

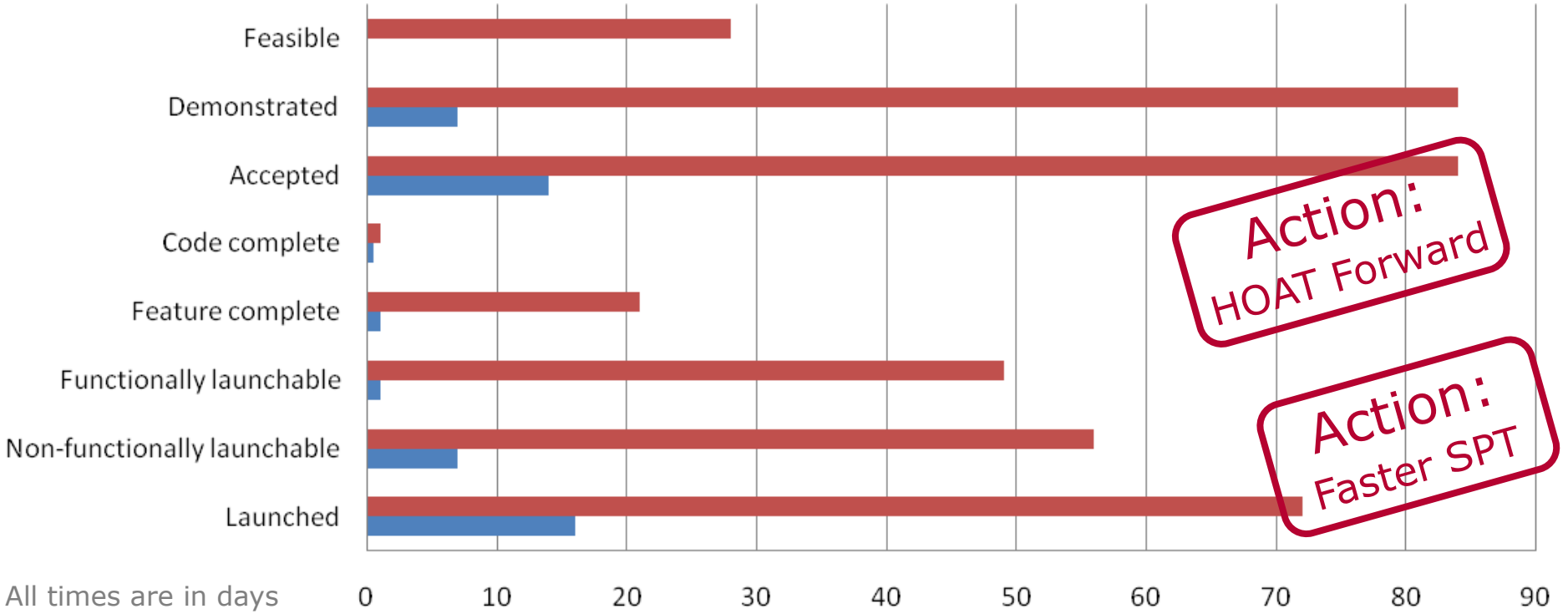
Faster Feedback
Eight Standard Measures



MAERSK LINE

Faster Feedback

Comparing GCSS with the X-leap on the Eight Measures



	Launched	Non-functionally launchable	Functionally launchable	Feature complete	Code complete	Accepted	Demonstrated	Feasible
GCSS	72	56	49	21	1	84	84	28
X-Leap	16	7	1	1	0,5	14	7	

WIP LIMIT of 8
on bottleneck

Daily standup

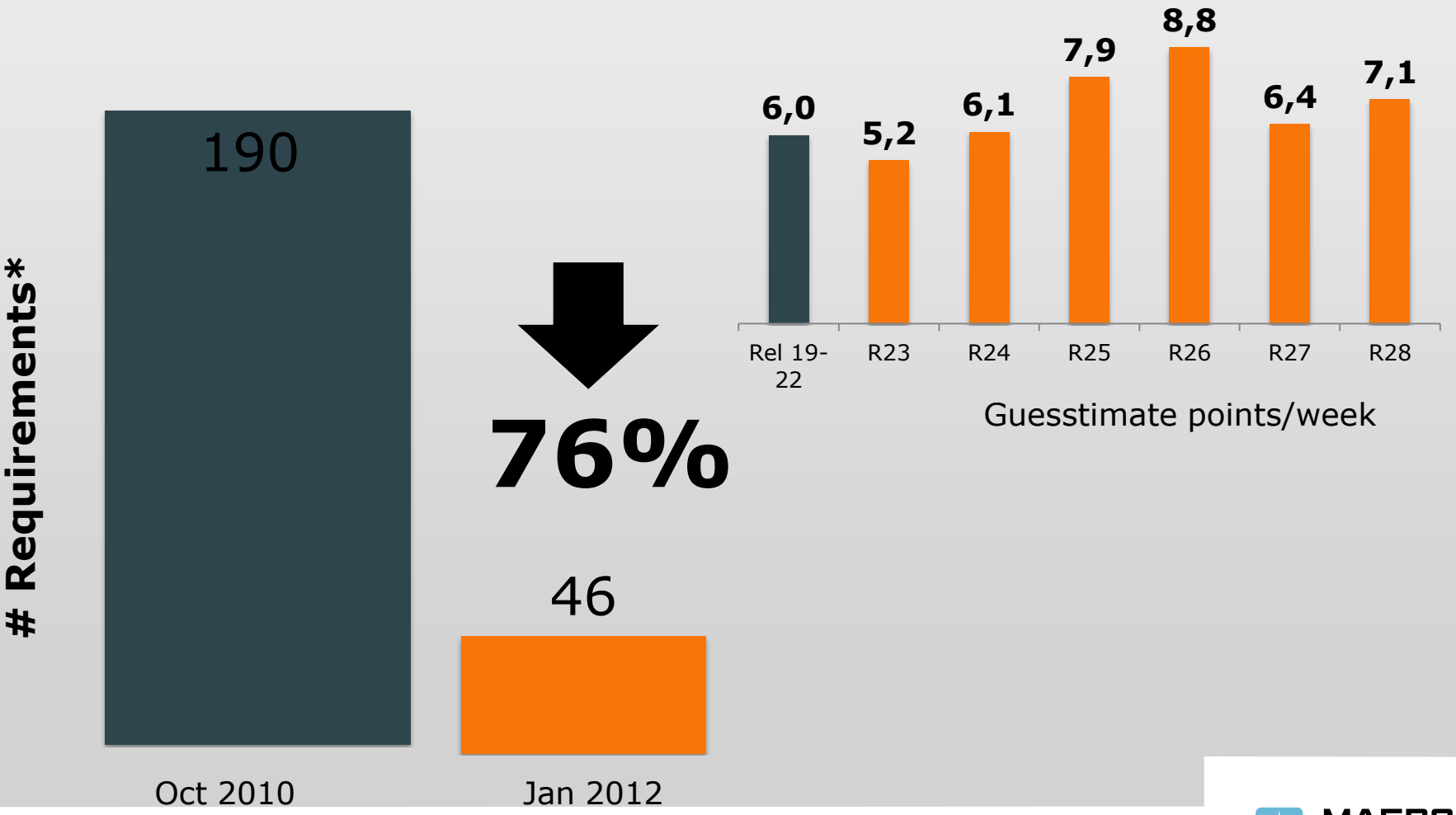
Changes to how code is handled

Streamline impact assessment

Improve error tracing

GCSS: Work-in-Progress reduced

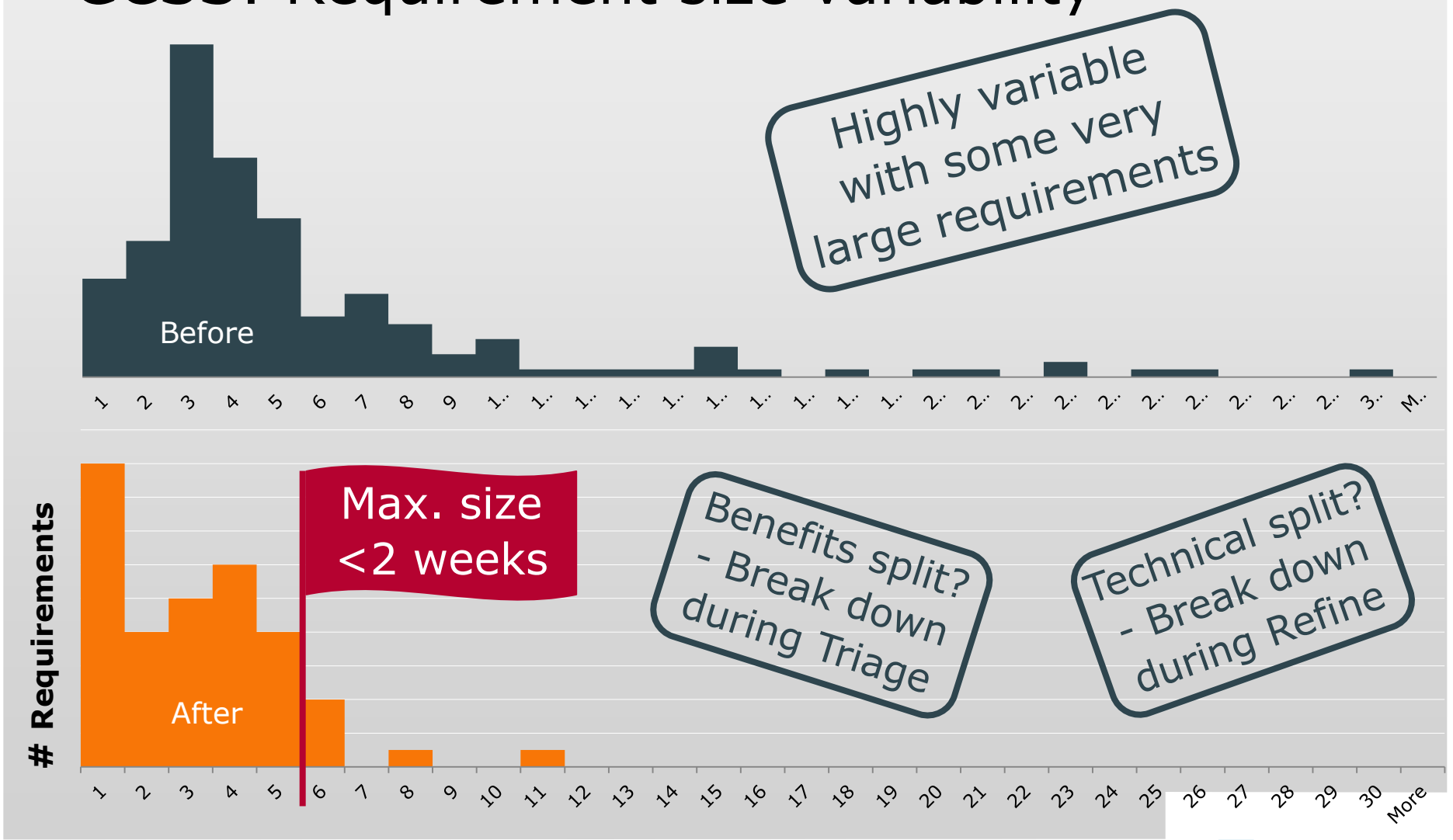
…whilst at least maintaining throughput



*"Authorized" to "Launched"

MAERSK LINE

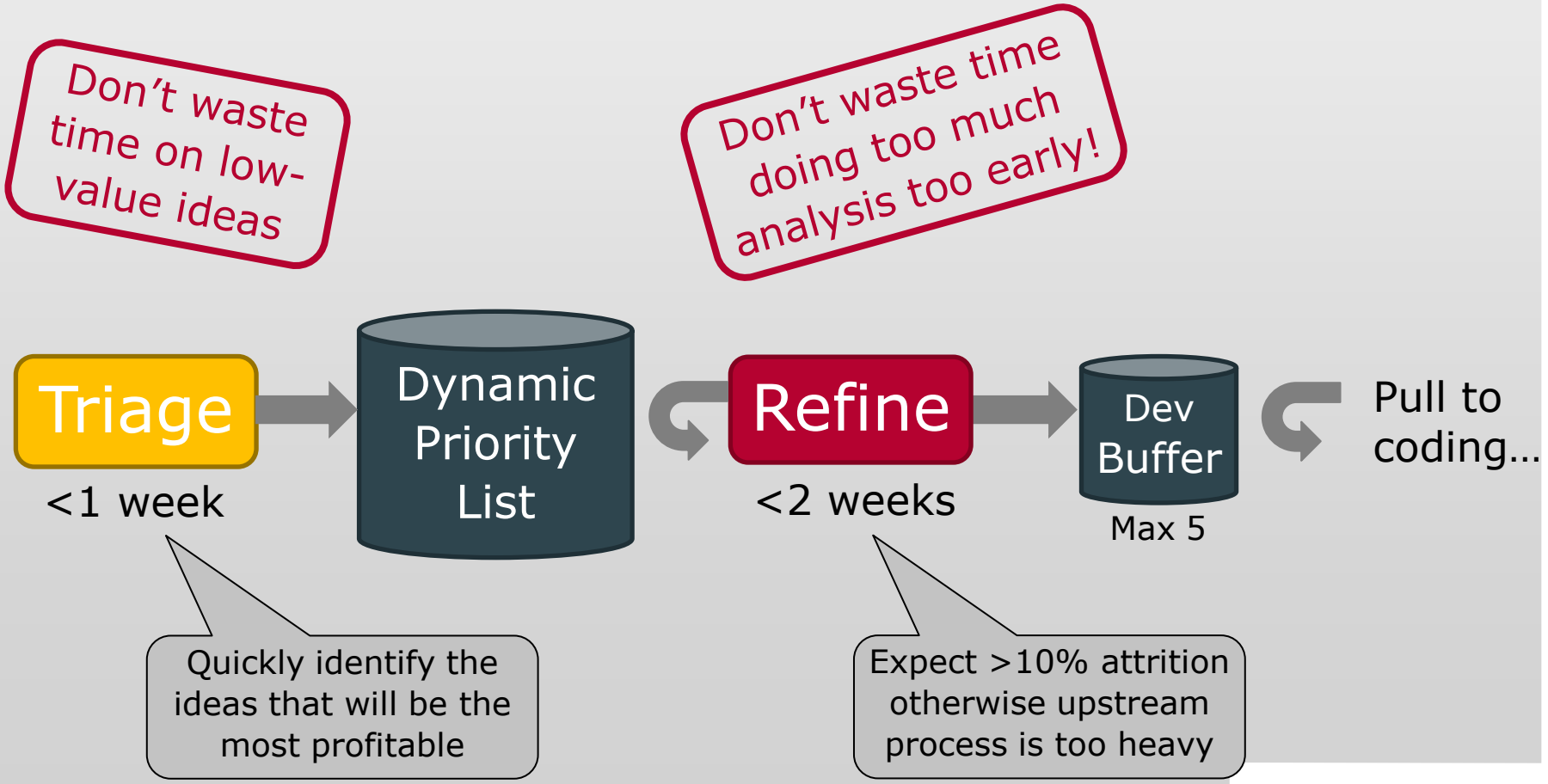
GCSS: Requirement size variability



MAERSK LINE

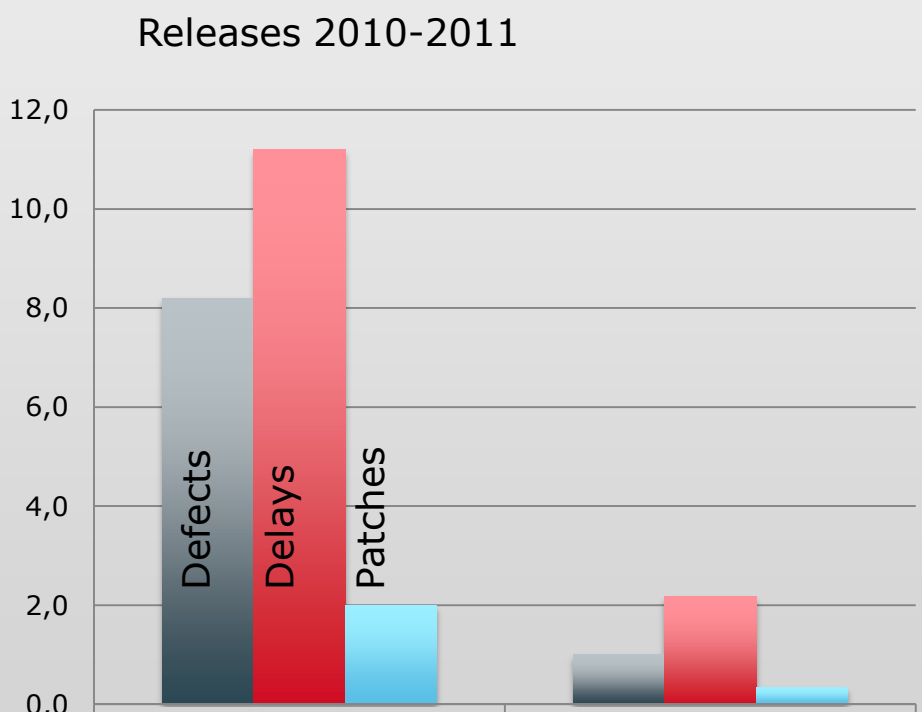
GCSS: Standardized Upstream Process

Get to initial prioritisation faster
Get to point of writing code quickly



MAERSK LINE

GCSS: Quality improvements



	Average Rel18-Rel22	Average Rel23-Rel28
E1+E2 Defects raised in HOAT	8,2	1,0
Production slippage (in days)	11,2	2,2
Patches 2wks after Prod	2,0	0,3

Up to June 2011 Since July 2011

-88%
Defects

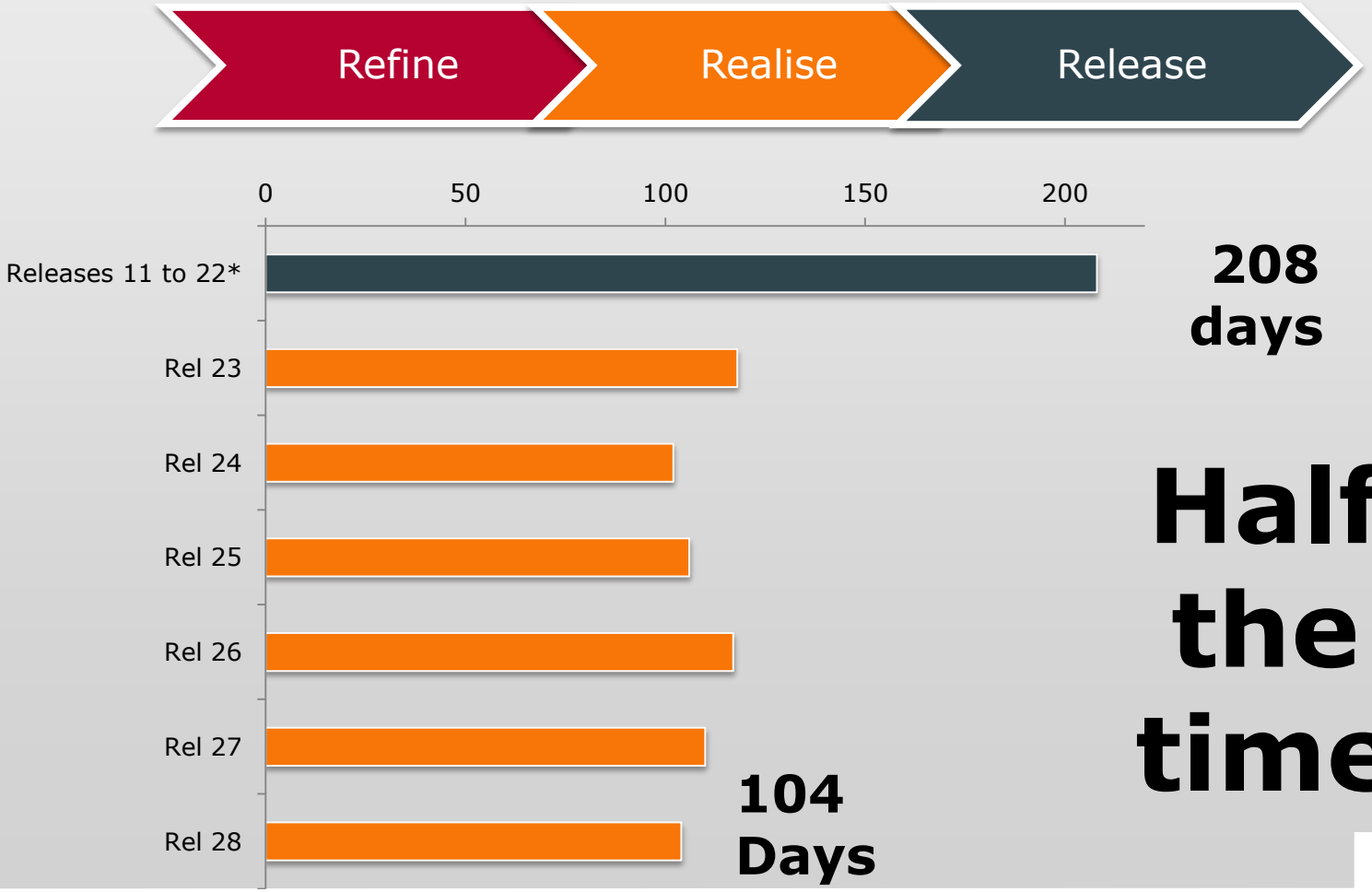
-80%
Delays

-85%
Patches

MAERSK LINE

GCSS: Cycle time

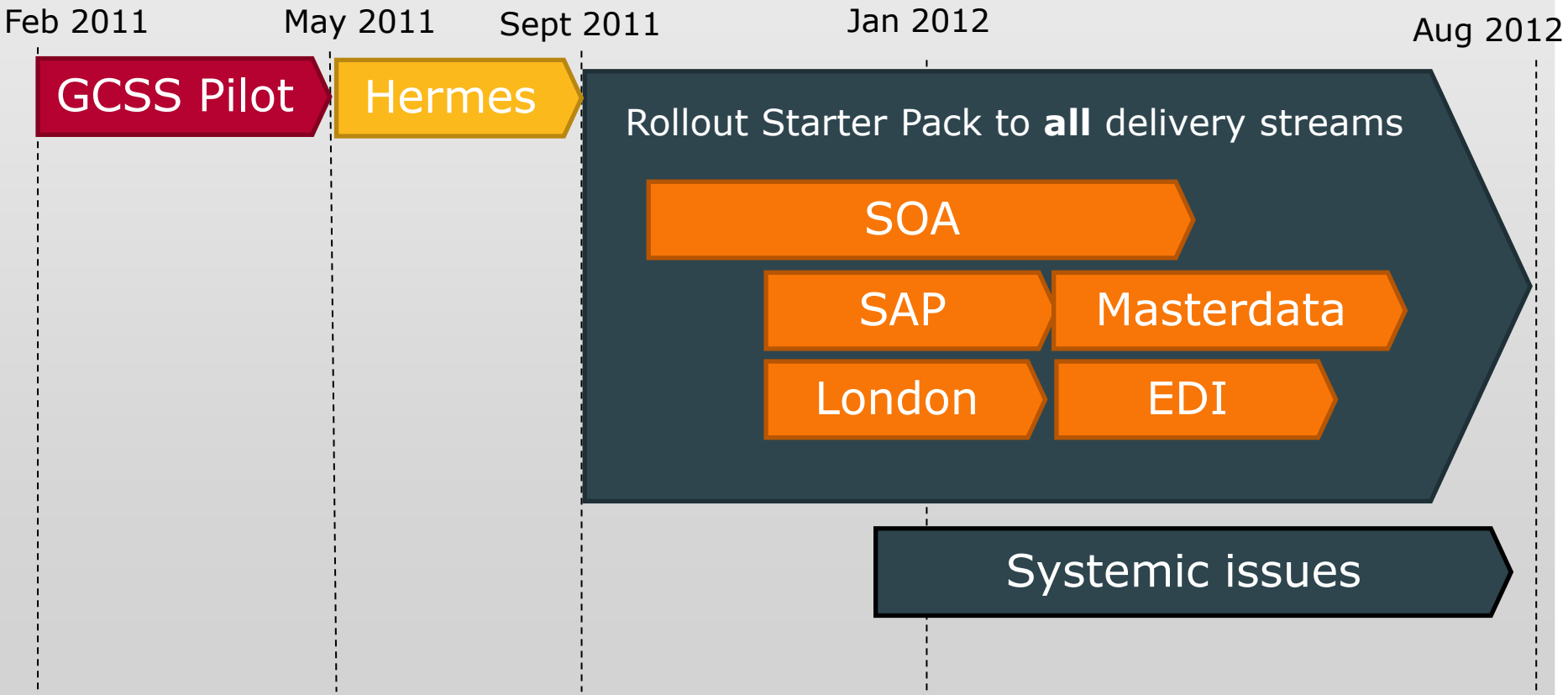
Average time elapsed from starting work to released



*No data for R18, R19

MAERSK LINE

Rolling out!



MAERSK LINE

Lean Product Development Checklist

MAERSK LINE

Start with these... The following are central to Lean Product Development

Bottom Line:

- [ ] **A working solution is delivered less than 90 days after starting work on a requirement** (when you've achieved this, go lower. For new projects, aim for less than 30 days)

- [ ] **New requirements are prioritised within a week of being captured**
- [ ] **Each requirement is supported by its own $ benefits assessment**
- [ ] **Each requirement is prioritised using Cost of Delay Divided by Duration (CD3)**

- [ ] **Requirements are pulled from a dynamically prioritised list when the development team have capacity**
- [ ] **The estimated size of a requirement is such that coding can be completed within two weeks**
- [ ] **The team starts writing code within two weeks of pulling a requirement from the dynamically prioritised list**
- [ ] **The team actively manages the number of requirements being worked on in at least one part of the process**

- [ ] **Developers get immediate feedback about whether changes they are making work or not**
- [ ] **When a requirement is completed it is demonstrated to the BPO for feedback and acceptance within a week**
- [ ] **Feedback about whether a release candidate is launchable is obtained within a week of code completion**
- [ ] **The team have regular Retrospectives where they capture learnings and drive improvements in how they work**

Then go further with these... If you are a new project, try to incorporate as many of these as possible from the beginning, *especially* the quality items

- [ ] Projects or major new enhancements have a Vision Document that describes the high-level purpose and scope
- [ ] All requirements are expressed as "User Stories" that form a placeholder for a conversation
- [ ] Someone with business knowledge who represents the customer is always available to the team for ad-hoc direction
- [ ] Risky items are prioritised by recognising the information value that completing them will generate

- [ ] The team sits together
- [ ] The solution is delivered in either small increments or short iterations *see other side
- [ ] The team have a daily review of progress where impediments are identified
- [ ] The team is made up of people with all the skills required to develop a solution
- [ ] Requirements have defined Acceptance Criteria before development starts

- [ ] Test coverage and code quality metrics are continuously monitored
- [ ] The design is simple to start with and evolves as new functionality is added
- [ ] All changes to the code base are immediately tested and reported on
- [ ] Developers work together in pairs and there is collective code ownership
- [ ] Executable tests are written before the solution is developed in code

Value — Quickly focus on the most profitable ideas

Flow — Smooth sustainable and just-in-time

Quality — Fast Feedback and flexible design

Engineering Quality Checklist

New delivery teams need to adopt these as soon as possible in order to build quality in and establish a foundation for sustainable delivery of value.

Development

- [ ] All assets are checked into a single repository (code, config., test scripts, schemas, migration scripts etc) 1
- [ ] Developers check-in code to the repository at least daily 2
- [ ] Developers have collective code ownership & responsibility 3
- [ ] Non-functional requirements are identified and prioritised alongside other requirements 4
- [ ] User interface tests & unit tests are run by the developer before code check-in 5
- [ ] Source control branches are frequently merged (every 2 weeks or less) 6

Technical debt

- [ ] The team regularly takes time to identify and record technical debt 7
- [ ] Repaying technical debt is prioritized alongside other requirements 8

Build & test

- [ ] The build runs all unit tests, regression tests and all non-manual acceptance tests 9
- [ ] Broken builds are fixed (or the check-in is reverted) before more code is checked-in 10
- [ ] Test stubs ensure all automated tests are independent of other systems (excl. network & integration tests) 11
- [ ] A build is completed within 20 mins of code check-in and is then deployed to a non-production environment 12
- [ ] Test coverage and code quality metrics are monitored 13
- [ ] Testing is prioritised using a risk-based approach 14
- [ ] Some performance tests are run at least daily 15
- [ ] The load-to-failure threshold is identified 16
- [ ] All programmatic interfaces are permanently available to other systems for integration testing 17

Deployment

- [ ] All batch testing of requirements and the subsequent deployment to production takes 7 days or less 18
- [ ] All environments can be recreated using the same automated process 19
- [ ] Updates are deployed to production without customer downtime 20
- [ ] All deployments are automated (including schemas, migrations & platform/application configuration) 21
- [ ] A developer's environment & tools are built from a standard configuration within 2 hours 22

Environment provisioning

- [ ] Any new environments (excluding production) required are provisioned within a week 23
- [ ] Any standard production environments required are provisioned within a month 24

Monitoring & improvement

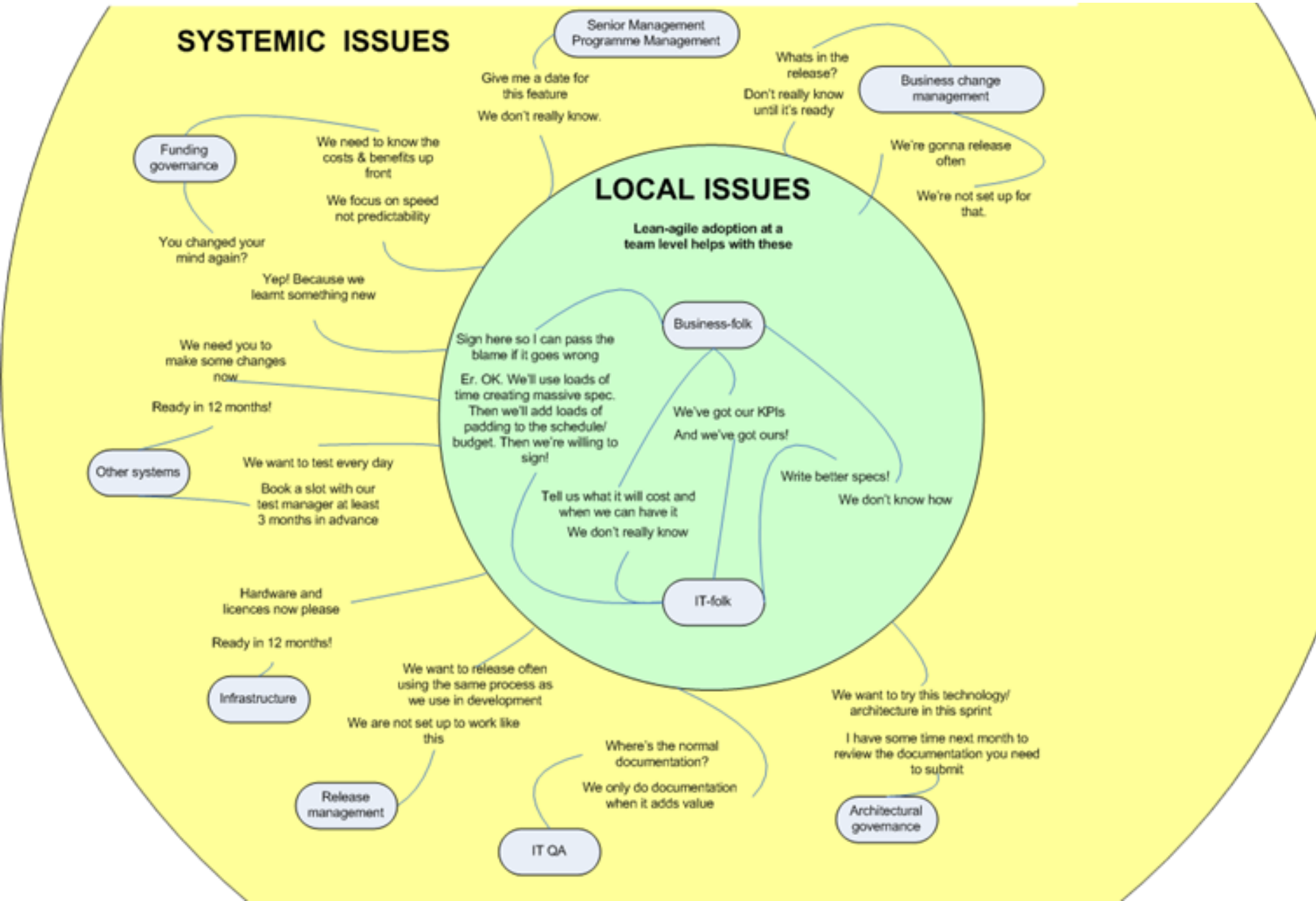
- [ ] Build, test & deployment process performance is measured and continually improved upon 25
- [ ] How to monitor production health is an integral part of the design 26

v1.0
12-2-2012

Learning from rollout so far

- Practices seem to work everywhere
- Mature teams are generally more receptive than newer ones
  - The know their process and that it needs improvement
- As with all change programmes, a couple of key individuals in the team can make a huge difference
- Personnel turnover make changes hard to stick
- There are systemic issues which need addressing

Potential lean-agile adoption barriers



Slow burn - stakeholder education

MAERSK LINE

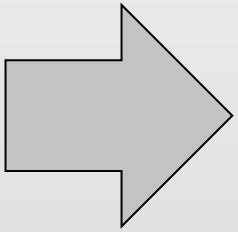
Key Performance Measures for IT

MAERSK LINE

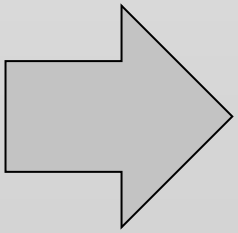
Variable	Typical measures	Usual outcomes	Alternative measures
Time	Delivering on a predicted date	Incentivises hidden time buffers and slower delivery	Maximise speed in getting to the point where value starts to be realised
Scope	Delivering all of the originally predicted scope	Incentivises gold plating and discourages exploitation of learning.	Minimize size of work packages to maximize both learning and early release of value
Cost	Delivering at or below a predicted development cost	Incentivises hidden cost contingencies, pushing costs up.	Maximize value delivered (trade development cost against the opportunity cost of delay)
Quality	Delivering changes with zero downtime and no errors	Resistance to making any changes. Overinvestment in testing & documentation.	Shorten feedback cycles at many levels (coding, defects…)

What next for Maersk Line?

- **Legacy:** Complete rollout 8 starter pack practices for all legacy applications

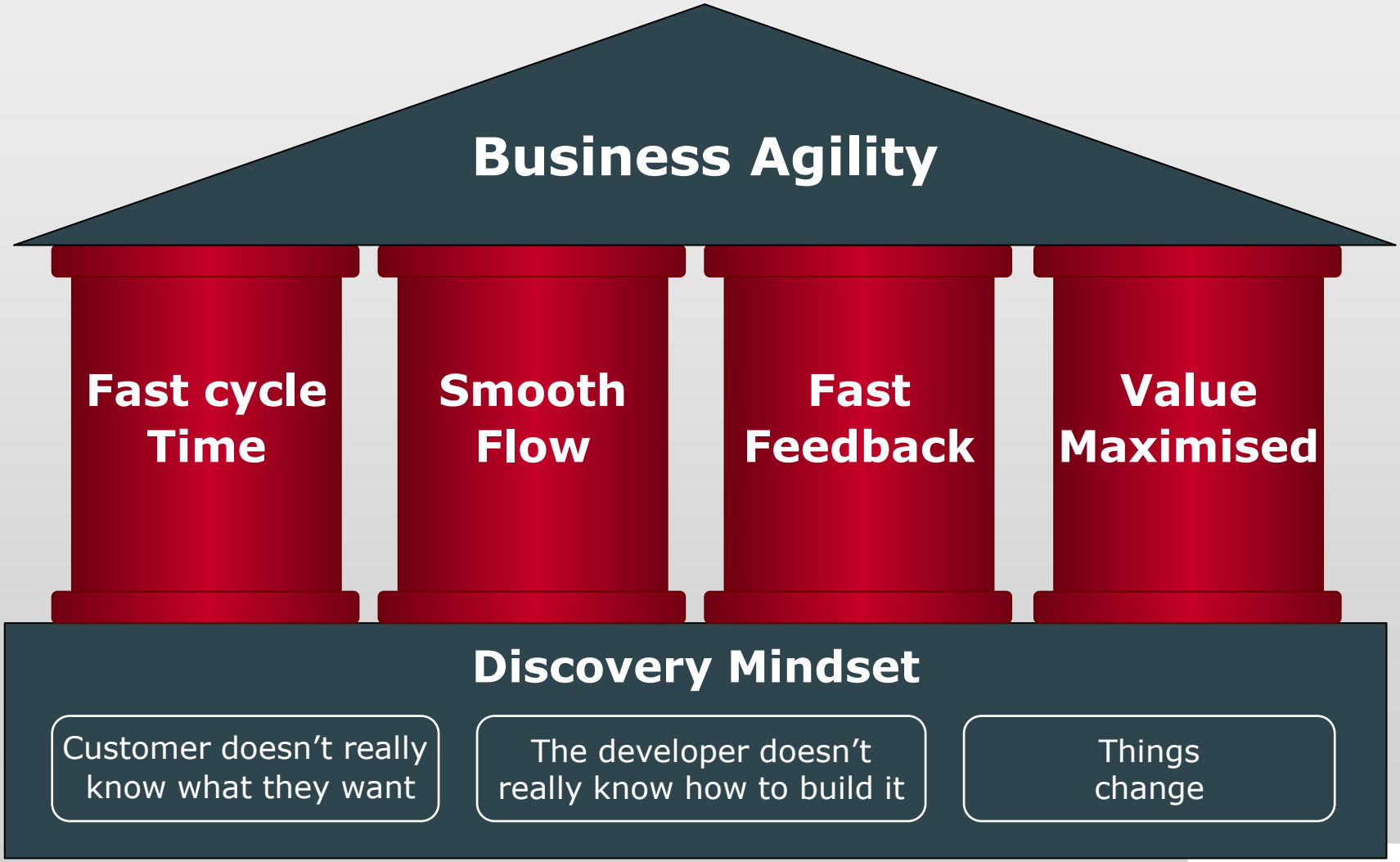


- **New:** Additional practices for our new Service Oriented ”vision platform”



MAERSK LINE

Enabling Agility



MAERSK LINE

Questions?

Chris Berridge
Programme Manager
Lean Product Development
Maersk Line IT

+45 3363 8165
chris.berridge@maersk.com

Agile Project/Programme Manager of the Year 2011

MAERSK LINE