Codethink

Providing Genius.

www.codethink.co.uk

Enough buzzwords!

Moving towards a more efficient and flexible delivery model in automotive environments.

Event: AGL AMM
Schedule:Sept. 7th 2016

Agustín Benito Bethencourt
Principal Consultant - FOSS

Who are Codethink?

- Provide software engineering & consultancy services.
- Expert in Linux and Open Source software.
- Focus on automotive industry, embedded devices & enterprise infrastructure.
- UK Headquarters, serving clients in EU, US and Asia.
- Independent and unbiased.
- AGL Bronze member.

Codethink Providing Genius.

Why Codethink?

Open Source company.

+

Automotive experience.

+

Passionate about building & delivering complex Linux systems.

Codethink Providing Genius.

The speaker: @toscalix

- Principal Consultant - FOSS at Codethink Ltd
- Experienced in managing people & programs/projects in the open at Codethink, Linaro, SUSE, ASOLIF...
- Involved in CIP/AGL (Linux Foundation) & GENIVI.
- More about myself at http://www.toscalix.com

Codethink Providing Genius.

Today, automotive projects suffer from...

1. Delivery processes are too complex.
2. Long stabilization phase.
3. Long lead times.
4. Delivering Open Source without taking advantage of how FOSS is developed.

Many more issues but… not enough time to cover them.

Providing Genius.

… in consequence...

… unbalanced effort in product delivery vs. software development.

1.- Too complex delivery process

Now

- Few, isolated, sequential, complex stages.
- Binaries over source code.
- Development is “far away” from delivery.

Effect

- Some delivery stages become systemic bottlenecks.
- Lack of delivery culture among developers & vice versa.
- Too early in the delivery process outcomes are hard to manage.

New pattern

- Source code over binaries.
- Integration as sync point, not release schedule or code.
- Many, interconnected, transparent, parallel and simple staged steps.

Key actions

- Automation to reduce effort… only once processes have proven to be efficient.
- Parallelization.
- Shared and managed software catalogue.

Codethink Providing Genius.

2.- Long Stabilization Phase

Now

- Long discrete production cycles.
- Little trust in deliverables by customers.
- Testing late, mostly done by testers.

Effect

- Long feedback loop.
- Defects found late. Too complex to analyze in depth.
- Severe changes needed during stabilization: back to production.
- Feedback from customers against “old but moving” requirements.

New pattern

- Treat test like you treat code.
- Developer, if you do not control the testing, you do not control your code.
- From a few complex long tests to many short and simple ones.
- Testing is part of everybody’s job.

Key actions

- Tests catalogue.
- Policies: Lego approach… like patches.
- Prioritize those tests that reduce feedback loop with developers.
- Share -> Efficiency -> Automation.

Codethink Providing Genius.

3.- Long Lead Time

Now

- Lead time = weeks/months.
- Complex roadmap design.
- Start from scratch every new product instead of product evolution.

Effect

- Long feedback loops between devs, integrators and testers.
- Low predictability and low flexibility.
- Long time to market so deadlines under too much pressure

New pattern

- Transparency.
- Integration as the meeting point: not code or release schedules.
- Reducing lead times is about reducing feedback loops.
- It is not about fixing bottlenecks but deprecate them.

Key actions

- Prioritise what reduces the feedback loops.
- Shared integration processes/team across the whole organization.
- Rolling delivery model for your base system.
- Walking (exo)skeleton first.

Providing Genius.

4.- Take advantage of how FOSS is developed (i)

- Apply Open Source principles internally:
  - Transparency within the organization.
  - Quality is everybody’s responsibility.
  - Work like those who develop the code you ship: work like upstream.
- Use popular Open Source development tools:
  - Git: become a source code version control/management expert.
  - Open Source CI and Code Review tools: your code is worth nothing in isolation.
  - Testing: distribute everything, the lab, the tests, the results, the analysis...
  - Bug tracking, MLs, IRC: what works for communities… works for you.

Providing Genius.

4.- Take advantage of how FOSS is developed (ii)

- Follow Open Source development best practices:
  - Code review: quality ≠ testing.
  - Dogfooding: are you using Windows while shipping Linux? Really?
  - Code and communicate as if you are in the open.
- Follow Open Source delivery best practices:
  - Rolling for moving fast. Branching for stabilization only. Master first.
  - Beta testing: no test can substitute a beta tester’s feedback.
  - Integration issues are also developer’s responsibility.
  - Testing is everybody’s topic. Treat tests like code.

Codethink Providing Genius.

As a result...

- Lead time and stabilization phase reduction:
  - Gain control over your products.
  - Reduce time to market and maintenance costs.
  - Learn faster. Earn trust from customers.
- Simplify your delivery process
  - Reduce integration risks: more but simpler issues.
  - Traceability / Reproducibility / Software re-usage.
  - Ease the management of providers' software.
  - Gain flexibility and predictability.
- Balance delivery vs. development effort:
  - Less focus on shipping the product and more on differentiation.
  - Delivery as a first class discipline.

Providing Genius.

In summary, at Codethink we believe that...

1.- It is not about doing more but about doing the right things… right.

In summary, at Codethink we believe that...

2.- There is no control without the source code AND the knowledge associated with it:

work with upstream

Codethink Providing Genius.

In summary, at Codethink we believe that...

3.- Once you gain efficiency and flexibility, delivery becomes a differentiation factor

Providing Genius.

In summary, at Codethink we believe that...

4.- If you consume, develop and deliver Open Source software…

… become an Open Source company

Providing Genius.

Thanks.

Questions?

Codethink Providing Genius.

www.codethink.co.uk