SIEMENS
Ingenuity for life

Our CI Journey

How our Firmware Team in Siemens Mobility moved to Continuous Integration

On your marks, …

SIEMENS
Ingenuity for life

Pre - Jan 2017

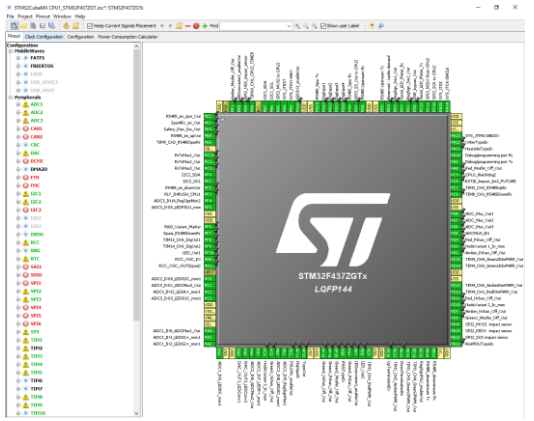
- Took an age to
  - Capture Requirements
  - Make and sell the Business Case
- Evaluated QI tools, selected Lattix, Klocwork, VectorCAST

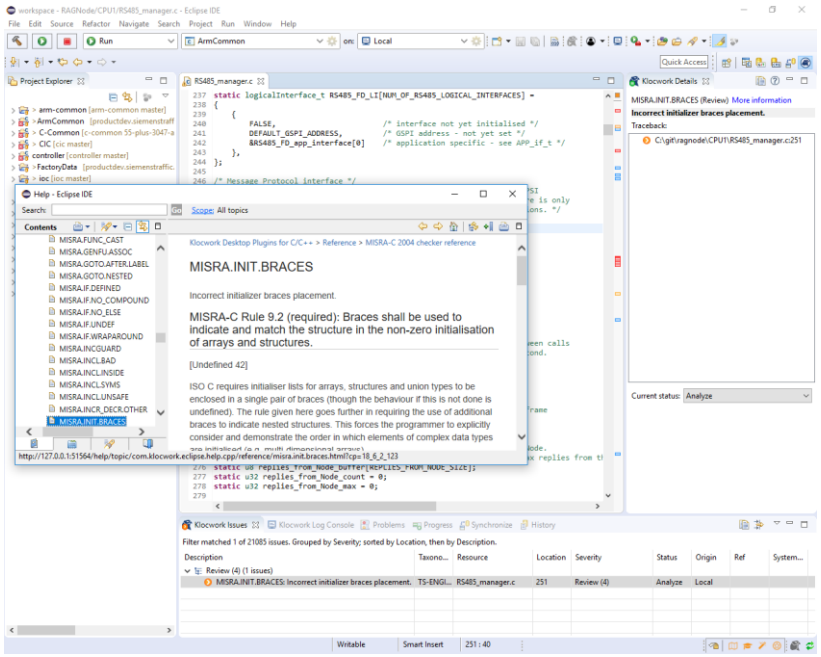
On your marks, get set, … Code !!

SIEMENS
Ingenuity for life

Jan 2017 – first cut of code - STM32Cube

- Embedded targets generated by STM32Cube
- Fortnightly Sprints, end-of-sprint builds (2 day)
- Eclipse IDE, CVS, Keil and Klocwork Plugins
- Use of changesets, post-checkin reviews
- *Push back from engineers on Quality Tools*





On your marks, get set, … Code !!

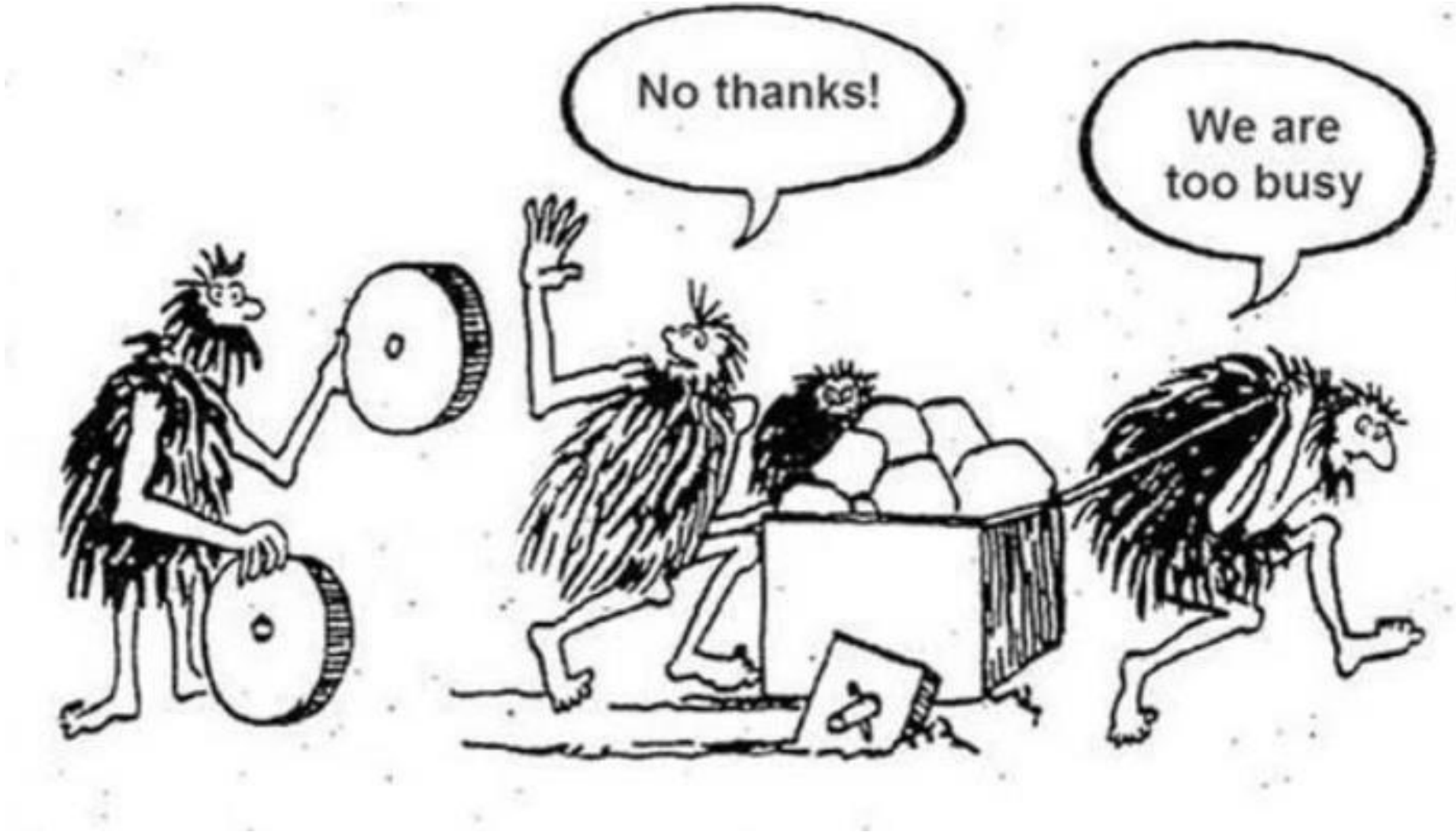
SIEMENS
Ingenuity for life

Jan 2017 – first cut of code - STM32Cube

- Embedded targets generated by STM32Cube
- Fortnightly Sprints, end-of-sprint builds (2 day)
- Eclipse IDE, CVS, Keil and Klocwork Plugins
- Use of changesets, post-checkin reviews
- *Push back from engineers on Quality Tools*

Sep 2017 – first Build – 1 month bring up

The Problem of Technical Debt



Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

By integrating regularly, you can detect errors quickly, and locate them more easily.

Source: https://www.thoughtworks.com/continuous-integration

On your marks, get set, … have a practise

SIEMENS
Ingenuity for life

Feb 2018 – Verifa trial via Jenkins CVS plugin

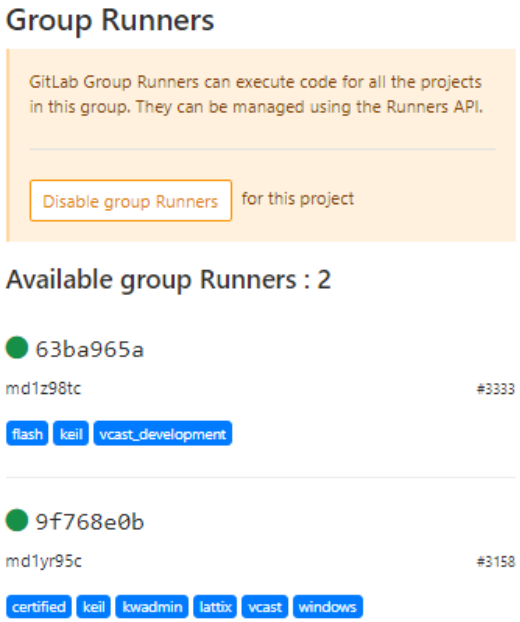
- Proof of concept to show value of CI
- Attempt to automate VectorCAST testing
- Bug detection – bug found on first pipeline run
- Proved need to automatically integrate and test software much more regularly

On your marks, get set, … Continuously Integrate !!

SIEMENS
Ingenuity for life

July 2018 – we take the plunge

- Repos move to code.siemens (Github based)



Why

We received several requests by developers to support them with ready to use answers to questions form their management about why their units shall sponsor and support the use of code.siemens.com. Here are some arguments you can use:

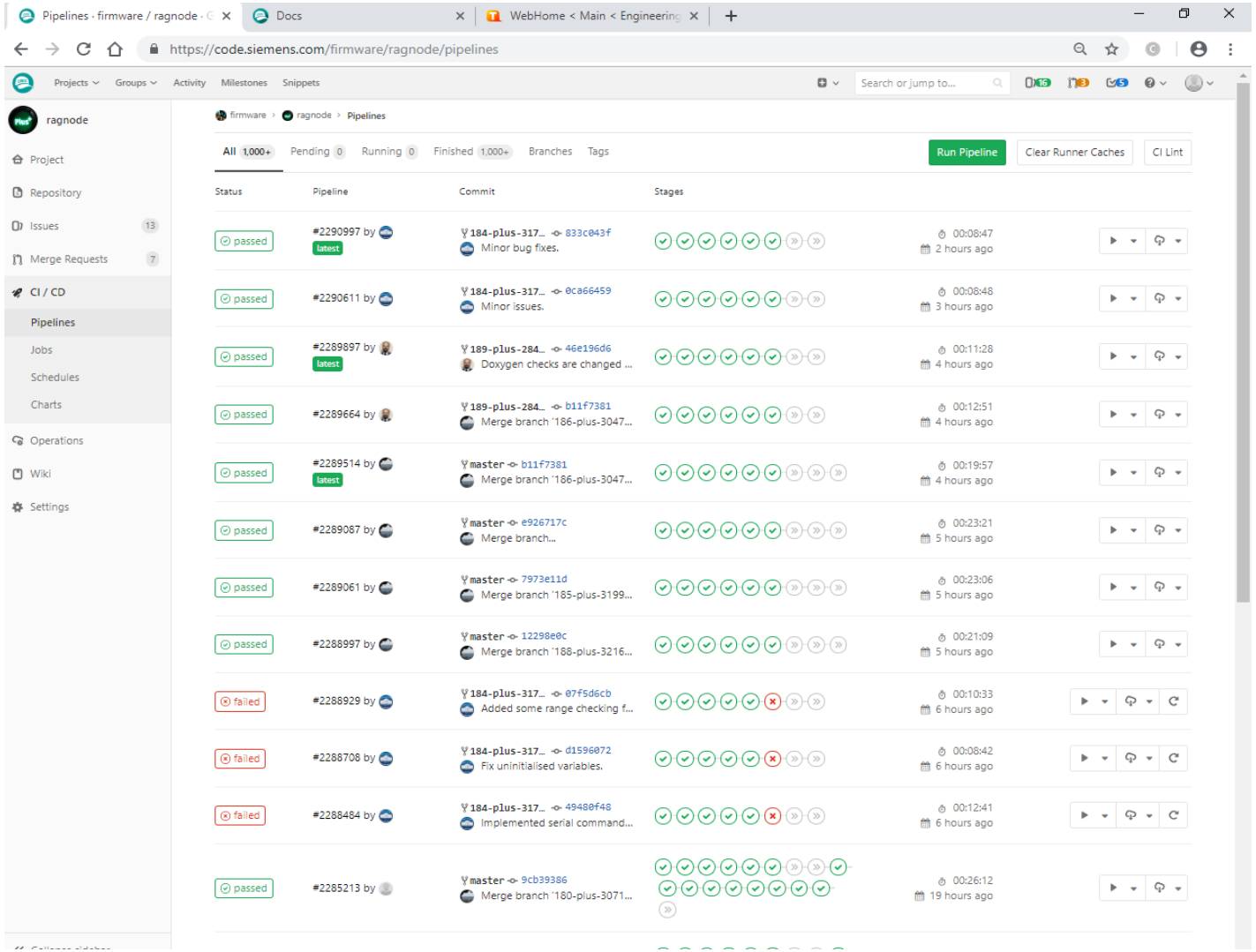
- collaboration across Siemens
  - one platform instead of many islands
  - best practice sharing on coding and ci/cd
  - not a platform only, it's a community with > 20000 users
  - use for open, inner and closed source up to ACP 2-2-2
- security
  - pentest every year
  - protection concept
  - ACP 2-2-2 rating
- operations
  - managed by pros
  - scales up
- cost savings
  - 67€/user per month was the estimate for ACP3-3-3 GitLab
  - open source based, no license fees
  - typical cost structure: license fees, compute and storage, network, operations, security, monitoring, support
- one stop shop
  - more than source code repository
  - issue tracking
  - continuous integration / delivery
  - APIs to automate further
  - docker.siemens.com
  - shared build runners included
  - option to use private runners

Continuously Integrate – Verifa Consultancy, Phase 1

SIEMENS
Ingenuity for life

July 2018 – Verifa Consultancy, Phase 1

- Repos move to code.siemens (Github based)
- Verifa consultant on-site for one week
- CI Pipeline-on-Push, build+klocwork+Doxygen
- CI Pipeline-on-Merge, build+klocwork upload



Continuously Integrate – Verifa Consultancy, Phase 1

SIEMENS
Ingenuity for life

July 2018 – Verifa Consultancy, Phase 1

- Repos move to code.siemens (Github based)
- Verifa consultant on-site for one week
- CI Pipeline-on-Push, build+klocwork+Doxygen
- CI Pipeline-on-Merge, build+klocwork upload
- CI Pipeline-on-Pipeline for 23 submodules

```
# Trigger RAG Node CPU1 build when master is updated
trigger-ragnode-cpu1:
  stage: trigger-ragnode
  only:
    - master
  tags:
    - AWS
    - DOCKER
  script:
    - curl --request POST --form token=[redacted] --form ref=master
                          --form "variables[MASTER_BUILD]=1" https://code.siemens.com/api/v4/projects/firmware%2Fragnode/trigger/pipeline
```

```
build-boot-master:
# Performs a build check on any master action, and also generates the klocwork build settings
# Variables set include:
# ADD_VALID_CHECKSUM: 1 - indicates that a valid checksum is to be added to the built file
  stage: build-boot
# Require runner to be Windows machine with Certified Keil Compiler
  tags:
    - windows
    - keil
    - certified
  only:
    - master
    - tags
# if the MASTER_BUILD variable is set, perform a git checkout master for each sub-module
# prior to running the script
  before_script:
    - git submodule status
    - if defined MASTER_BUILD ( echo Build with pipeline-support master HEAD &&
                                git submodule foreach "git checkout master && git pull origin master"
                                && git submodule status)
  script:
    - cd CPU2\CPU2_STM32F100V8\MDK-ARM\boot
    - call .\%PIPELINE_PATH%\init-environment.cmd
    - .\%PIPELINE_PATH%\build-uv4.cmd
  variables:
    PROJECT: "%PROJECT_BOOTLOADER%"
    PIPELINE_PATH: "..\\..\\..\\..\\pipeline_support"
    ADD_VALID_CHECKSUM: 1
  artifacts:
    # save built files to code.siemens.com
    # These include the klocwork settings file, which can be re-used in the subsequent klocwork-boot-master job
    # (note that both these jobs are guaranteed to run on the certified machine).
    # The Job artifacts can be downloaded from the build-boot-master Job page for formal testing
    # or when creating a release.
    when: on_success
    name: "%CI_JOB_NAME%-%CI_COMMIT_SHA%"
    paths:
    - CPU2\CPU2_STM32F100V8\MDK-ARM\boot\CPU2_STM32F100V8\CPU2_STM32F100V8_boot.map
    - CPU2\CPU2_STM32F100V8\MDK-ARM\boot\CPU2_STM32F100V8\CPU2_STM32F100V8_boot.hex
    - CPU2\CPU2_STM32F100V8\MDK-ARM\boot\CPU2_STM32F100V8\CPU2_STM32F100V8_boot.axf
    - CPU2\CPU2_STM32F100V8\MDK-ARM\boot\CPU2_STM32F100V8\CPU2_STM32F100V8_boot.build_log.htm
    - CPU2\CPU2_STM32F100V8\MDK-ARM\boot\kwinject_build.out
    expire_in: 1 week
```

Continuously Integrate – Verifa Consultancy, Phase 2

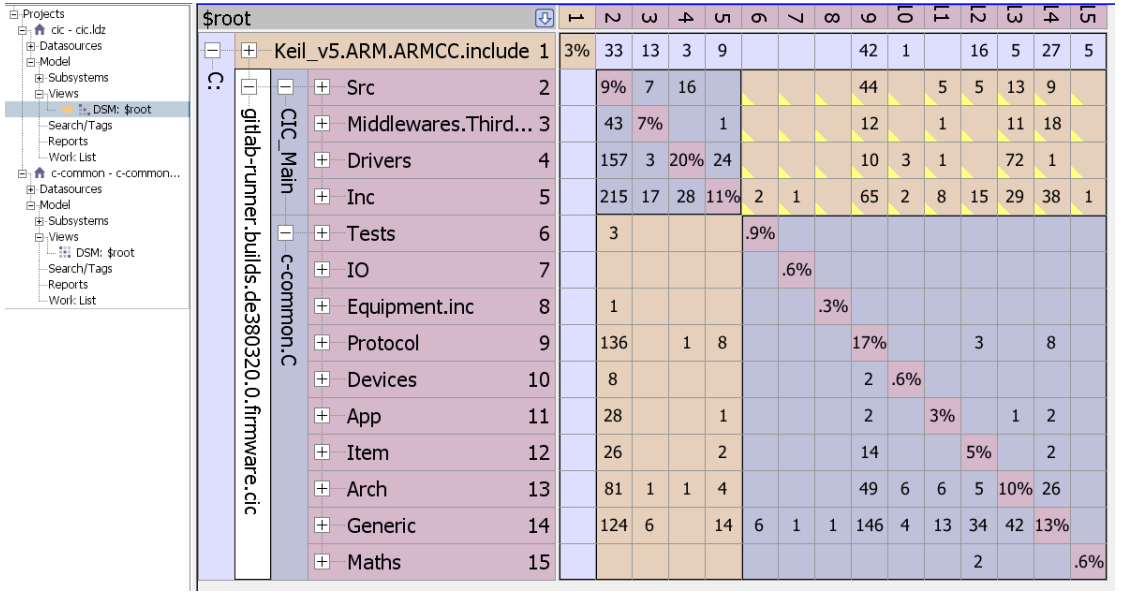
SIEMENS
Ingenuity for life

July 2018 – Verifa Consultancy, Phase 1

- Repos move to code.siemens (Github based)
- Verifa consultant on-site for one week
- CI Pipeline-on-Push, build+klocwork+Doxygen
- CI Pipeline-on-Merge, build+klocwork upload
- CI Pipeline-on-Pipeline for 23 submodules

Oct 2018 – Verifa Consultancy, Phase 2

- Improved Infrastructure – pipeline machine
- CI Pipeline Architectural Analysis, Lattix
- CI Pipeline L1 Unit Test, VectorCAST
- CI Pipeline Change-based testing (git diff)
- Scheduled pipelines for Unit Test
- *Push back from management on licences*
- Sprint test builds on demand (4 hours)



Unit	Subprogram	Complexity	Statements	Branches
application.c	init_system	6	27 / 31 (87%)	14 / 18 (77%)
	init_peripherals	2	9 / 9 (100%)	3 / 3 (100%)
	GSPI_generic_systemReset	2	2 / 2 (100%)	3 / 3 (100%)
	GSPI_generic_setLowPower	1	2 / 2 (100%)	1 / 1 (100%)
	GSPI_generic_getLowPower	1	1 / 1 (100%)	1 / 1 (100%)
	init_software	2	47 / 47 (100%)	3 / 3 (100%)
	init_modules	6	13 / 13 (100%)	11 / 11 (100%)
	application_main	1	1 / 1 (100%)	1 / 1 (100%)
	softwareErrorAppCallback	1	1 / 1 (100%)	1 / 1 (100%)
	productionTestControl	1	1 / 1 (100%)	1 / 1 (100%)
	getProductionTestStatus	1	1 / 1 (100%)	2 / 3 (66%)
	APPLICATION_get_address	1	1 / 1 (100%)	1 / 1 (100%)
	APPLICATION_get_wd_faults	5	8 / 8 (100%)	9 / 9 (100%)

Verifa Consultancy helped get things rolling

SIEMENS
Ingenuity for life

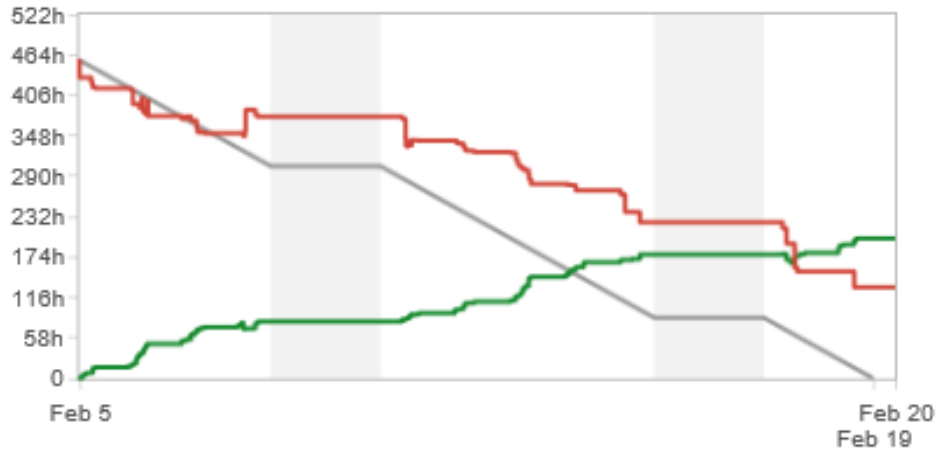
Allowed engineers to continue “adding value”

First month - £10K

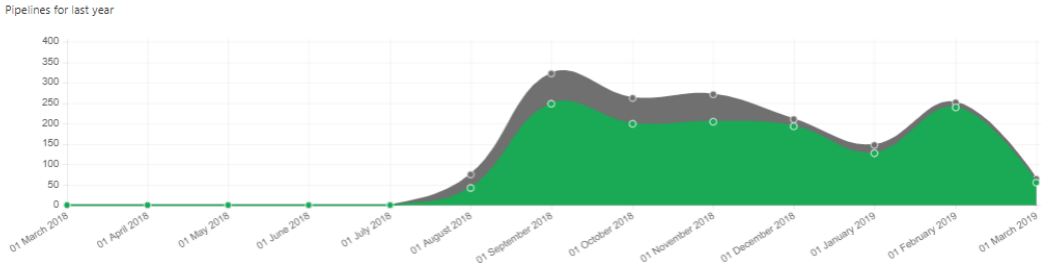
- Active projects moved from CVS to code.siemens, with use of pipelines from day 1
- Engineers now work on multiple branches
- Integration of existing tooling, e.g. Keil uv4, Klocwork, Doxygen, gave confidence
- Automated invocation of application pipeline on sub-module merge, re-use across products

Second month - £11K

- Architectural compliance to future proof
- Automated Unit Target testing in pipeline
- Scheduled pipelines defined
- Siemens pipeline available via choco install
- *Tools licensing issues caused delays*



Status Report



Continuously Innovate – engineers go off-piste

SIEMENS
Ingenuity for life

Dec 2018 – Release scripting

- Release script in tools submodule

Jan 2019 – Top level build in AWS Docker

- Sprint test builds on demand (3 hours)
- Definition of Unit Tests for Safety Critical code

Feb 2019 – EMC Test build

- No test failures

Tip: There are many ways to interact with the code base. Tortoise Git, Git for Windows, Eclipse EGit, all have their advocates …

Final Comments

SIEMENS
Ingenuity for life

"I only realised I didn't have the Klocwork plugin set up on Eclipse because the pipeline failed"
"Separate work on branches"
"Review & resolve actions before merge"
"Visibility of history is clear: branches & changes & differences"
"Standard products (git & GitLab) so much help available on the internet"
"Checks of pushes/merges done by pipelines"
"Build artefacts available from pipelines"
"Introduced quickly & easily into ongoing project"
"Improved framework for managing codebase"
"Eclipse seems to have poor support for git, particularly sub-modules. Recommend using command line."

David Hoslett
Senior Firmware Engineer
Siemens Mobility, Intelligent Traffic Solutions

Sopers Lane
Poole

Phone: +44 1202 782085
E-mail:
david.hoslett@siemens.com

siemens.com