GDP retrospective: 2015Q4 and 2016

Summary

This is the report done by the Codethink Ltd team that have been part of the GDP Delivery Team, taking care of the GDP project from September 2015 up to November 14th 2016. This report has two goals:

- Provide an overview of the status of the project
- Provide information to the new GDP delivery Team about the topics that remain open and the key decisions that need to be taken in opinion of the former team, based on the goals that have been driving the project so far.

The report is divided in three sections:

1. On the left column you will find the most important actions and decisions taken by the former GDP Team.
2. On the right side the team has described the key topics that are open at this point together with some of the opinions or lessons learnt that can help the coming team to approach them.
3. The third part provides some conclusions and opinions about the project as such or key topics.

Note: the coming sections are meant to be read from left to right so for each section you read the most relevant closed actions and the open ones.

Closed actions

Outcomes

Master is the main outcome for contributors to the project and other automotive system developers interested in Open Source who have Yocto knowledge. Master is where collaboration at GENIVI takes place. You can find there the latest automotive software available, including the latest software developed by the GENIVI Alliance. Master allows them to build from scratch images for their favourite target and customise them. GENIVI Master was created in May 2016. It is a rolling release based on three simple principles:

1. There is only one repository where all the code is integrated.
2. That repository is always working, always.
3. Every patch/code goes to Master first, then to branches.

GENIVI Development Platform (GDP) is a distribution derivative from poky, build with bitbake (Yocto), that includes the GENIVI Alliance developed automotive software components as part of the platform for the industry, presented as binaries, with ports to a variety of popular development boards. It has had one major release in 2016. You can find a more detailed description of these outcomes in:

- The GENIVI wiki
- The GDP Summary presentation, delivered at the 15th AMM.

GDP maintainers has admin/write rights on the following repositories:

- genivi-dev-platform.git
- meta-genivi-dev.git
- gdp-test-suite.git

Open actions

Outcomes

Actions to be performed for any of the outcomes maintained by the GDP Delivery team, that is Master and GDP, are under the following Epics:

- GDP 11: tasks related with GDP 11 releases and pre-releases. Stabilization, deployment, publishing and marketing activities performed or coordinated by the team.
  GDP-273 IN PROGRESS
- Master: tasks related with Master. Since all the integration and fixing takes place in Master, you can find here most of the technical tasks since it was created.
  GDP-257 IN PROGRESS
- Management: all the management tasks, those related with contents, processes or any other that does not fit anywhere else, is under this Epic. **GDP-11** IN PROGRESS
- GDP-ivi9 is the first release done by the GDP Delivery Team. It is in maintenance until GDP 11 is release, so it is still open. **GDP-8** IN PROGRESS

The other epics under the GDP JIRA space are not managed by the GDP Delivery team. There is work to be done to update those tickets not managed by the GDP team and relate them with new GDP requests. In other words, more coordination between PMO and GDP Team related to the PMO tickets is needed.

GDP releases

GDP releases

The following releases have been done during the past year:

- GDP 11 releases:
  - GDP 11 RC3 support for RPi3 only. Integration and shipping of the new Application Launcher and demo apps. Release for the 15th AMM. GDP-414 DONE
  - GDP 11 RC2 support for QEMU, RPi2 & 3, Renesas Porter and Silk, Dragonboard 410c and Minnowboard MAX and Turbot. Improved the overall stabilization of the system. GDP-376 DONE . It was developed thinking this was going to be the second major release of 2016, before the new HMI came a few days before the planned release. It has the stability of a major release.
  - GDP 11 RC1 support for QEMU, RPi2 & 3 and Minnowboard MAX. Focus on system stabilization and ports creation. Support for the RPi2/3 official touchscreen was added. GDP-335 DONE
  - GDP 11 Beta: support for QEMU only. Moved from Yocto 1.8 to 2.1. Significant update effort. GDP-294 DONE
- GDP-ivi9 releases:
  - GDP-ivi9 final release. Still maintained (until GDP 11 release is out). Stabilization of the whole system was improved. GDP-36 DONE
  - GDP-ivi9 RC1. Release processes were improved. This is the first time we focused effort in creating ports to development boards. Stabilization improved. GDP-35 DONE
  - GDP-ivi9 Beta. This was the first release. We set up the basic release processes and did the first significant update to the base system from the previous version. GDP-17 DONE . It involved a significant effort in updating the base system from the inherited GDP platform from the previous provider.

You can find more information about each release in the GDP Releases wiki page. As result of releasing GDP, JIRA creates a list of completed tasks for each release that is being published in the GDP Release report: tasks completed wiki page. Wiki pages and attachments related with releases should have the label *release*.

GDP Master

- You can find a definition of Master in GDP FAQ.
- The description and a summary of the policies applied to Master can be found in the GDP Master wiki page.
- Maintenance policies can be found in the GDP Management wiki page.
- Master was created right after the 14th AMM. You can find a summary of the articles published about it in the GDP blog:
  - Technical article describing aspects of Master, written by Tom Pollard, published in June 2016.
  - Article describing the structure of the GDP outcomes now that Master was in place, written by Agustin Benito Bethencourt and published in June 2016.

The first main task for the GDP Team will be to release GDP 11. In order to do that, the following actions will be required:

- Define GDP 11 release schedule GDP-272 TO DO . The current release cycle has suffered several last minute changes that has interfered with the engineering work. Consolidating a release schedule is needed to achieve technical excellence, a requirement at this point for the project, in our opinion.
- Create GDP 11 release task + subtasks structure, based on GDP 11 RC2 and GDP-ivi9 releases structure, using JIRA features for this purpose: GDP-297 TO DO . This is key to planning the Release since the Release process is manual and labour intensive in several key areas. Take the following as examples:
  - GDP 11 RC2 release tasks: GDP-376 DONE
  - GDP-ivi9 release tasks: GDP-36 DONE

It needs to be decided how the new outcomes, SDE and QtAS relates to GDP when it comes to the release process. Are they going to be part all of the same release? Will they have different release timelines? Coordination with maintainers will be needed.

GDP Master

- Requests to be considered for Master releases are managed following a specific process.
  - Increase flexibility: GDP-448 ANALYSIS
    - Master is the default place for those who understand Yocto. Today GDP ships Master completely. This should not be the case in the future, as more code lands to Master. There should be a filter on what lands to GDP compared to what it lands to Master, which should be everything. In order for system developers to choose what software they want to include on the build, included in Master or not, more flexibility of how Master is built and structured should be improved.
    - The main goal is that customising builds should be not just allowed but promoted. By doing so, we increase the number of people using Master, so beta testers, and the number of use cases tested. Use cases that would never be tested by the GDP delivery team otherwise.
      - On the downside of this policy, the team would have to face the support requests for such cases, which is far from ideal, specially in a professional environment, with high board prices and usage of proprietary software, unaffordable for the Delivery Team. To reduce this impact, it is imperative to keep growing the community so contributors support newcomers.
  - Some of the latest requests are: GDP-431 ANALYSIS , GDP-405 ANALYSIS or GDP-224 ANALYSIS

        - White paper justifying the convenience of Master, written by Agustin Benito Bethencourt and published in July 2016.
    - Master requests are collected in a specific task: GDP-154 TO DO . Once accepted, each subtask is moved as task and split into subtasks if it is big enough to be done in steps. If it is related to a specific release, a FixVersion is added and a deadline. To know more about this topic check the specific section at the GDP management page.
    - An initial test infrastructure and some acceptance tests hav e been added. These are designed so that other tests can be added using the same pattern as the existing ones. These tests are written in Python and run on the QEMU build of GDP running the tests from the host machine which communicates with the virtual GDP using ssh. See the repository README for fuller instructions.

Content related with Master should have the label ***gdp_master,*** as stated in the label catalogue.

GDP ports to development boards

Currently the supported boards in Master are:

- Intel architecture
    - Minnowboard MAX
    - Minnowboard Turbot
    - QEMU X86-64
- ARM architecture
    - Raspberry Pi 2 and 3 (ARMv7 - v8)
    - Renesas Porter (ARMv7)
    - Renesas Silk (ARMv7)
    - Qualcomm DragonBoard 410c (ARMv8) Maintained by Changhyeok Bae

You can find links to the target board pages, with instructions on how to build port from Master for each board in GDP Releases wiki page. Every wiki page or attachment related with boards should have the label *target_board*. Please check the Management wiki page to find out more information about the policies related with the ports.

GDP Peripherals

Much work has been put into supporting peripherals for GDP. This has included:

- The faytech touchscreen. This device has needed two sets of kernel patches to get the touchscreen to work. GDP-233 DONE .
- The raspberry pi official touchscreen now works with GDP GDP-206 DONE .

Contents

The content structure for GDP is explained in the image published in

- Improve the stabilization of the new Application Launcher and demo applications.
    - This is needed as a pre-requisite for the GDP 11 release.
    - With a new HMI and new applications, given that there has been an intensive work on the platform itself and that the user base is increasing, there is an expectation of finding more bugs than before but with less impact on the system.
- The list of open issues in Master is published in the Open Bugs section of the Weekly Report page. Most of these are related to RC3 and need to be considered in the stabilization phase before the GDP 11 release.
- The acceptance test suite should be expanded to comprise further tests - some of which should interact with the HMI rather than just testing the base system. In order to utilise other tests the suite should be expanded so that tests developed externally maybe so that tests in languages other than Python can be run GDP-444 ANALYSIS .
- Get rid of unmaintained code in Master.
    - Unmaintained code has been a source of time waste for the GDP Delivery Team. This waste is increasing over time.
    - The more focus we put on the GENIVI components, the bigger the impact of unmaintained code in the daily work for the GDP Team..
- Limit PoC to Master for maturing.
    - GDP should only include components mature enough to be used by non-experts. Master is the place for maturing components.

GDP ports to development boards

- Dragonboard is currently being maintained by Changhyeok Bae. A decision to be taken is if the board will be supported by default by the GDP Delivery Team (official support).
    - Unless the investment in infrastructure and services increases, it is questionable that the GDP Delivery Team can increase the number of supported boards. The recommendation would be to reduce them, finding maintainers for some of the current ones.
    - If meta-genivi-dev can support a Renesas bsp-sublayer (as it does for intel + raspberrypi) it would help with the joint maintenance of the Renesas boards, and possibly make it easier to track the canonical meta-renesas upstream.
- Support *dd* without the need of sudo for Minnowboard images (description) GDP-370 TO DO .
- The QEMU port needs investigation with a possible switch to DRM GDP-166 TO DO the additional slowness caused by the new HMI also requires attention GDP-432 ANALYSIS .
- The boards pages - apart from the description and how to get it booted with a GDP image - only mention current problems, it would be good to add a 'features enabled in GDP' section.
- GDP 11 feature page can be improved adding more information about the features shipped in the coming major release.

Contents

- Finish GDP FAQ

the Release Howto wiki page. Also in that page is explained the content critical path and the status of it.

The most useful wiki pages for the maintainers to become familiar with are as follows:

- GDP Master
- GDP releases (and its subpages)- giving a summary of release content and instructions and advice for the supported boards.
- GDP Download -providing the users with access to prebuilt images
- The weekly reports - giving an insight as to why particular decisions were taken.
- GDP in detail - giving a breakdown for the component packages and software.

The repository GITHUB README pages also provide a useful way-in for new maintainers.

Users behaviours/analysis

Google analytics has been configured for the ftp server and Confluence. No significant actions has been taken to do any deep analysis of the data. In order to take conclusions, it is recommended to collect data during 3 months around January 2017. Some early conclusions can be extracted from a simple analysis though:

- The migration from the former to the new wiki has left broken links that has a significant impact at this point, leading users to Not Found pages. Jeremiah Foster and Nic holas Contino are aware of it and they are taking action, as reported during the Tools Team f2f meeting at the 15th AMM.
- GDP 11 RC2 and RC3 release announcements had only around 100 visits. Together with the fact that there are very few new users compared to returning ones that visit the GD P Download page indicates that there is a very long way to go on the marketing side.
- The GDP Download page is more visited than the landing page, which is a good sign. GDP is the most visited project.
- Checking the load page times, it is obvious that board images in target pages should be compressed. (link to tickets).

Service and infrastructure

JIRA

GDP project use JIRA as task management system and bug tracker. Bugs are easily identified because the bug issue type is used. The rest of the issue types are used for task management purposes. In general, very little customization has been done in JIRA. You can find information about how JIRA is configured in the GDP management page.

Task are structured through Epics 5 issues . The following Epics structure most of the GDP delivery team activity:

- Tasks related with GDP 11: GDP-273 IN PROGRESS
- GDP Master: tasks related with GDP Master: GDP-257 IN PROGRESS All the technical tickets are managed under this Epic unless they are referring to the
    - The more popular the project becomes, the more repeated questions we will get. It is important then to have the most basic ones answered on a FAQ. GDP FAQ should be completed with, at least, the open ones.
- Once GDP has shown that it is easy to deploy in your favourite target board, the focus should be put in the automotive components, which is what makes GENIVI unique. In order to do this, the critical content path should be extended, as shown in the picture, to each component. So each maintainer should create a wiki page for newbies, explaining in simple words what is the component for, how is integrated in Master/GDP, how to run it and some instructions that shows its capabilities..
    - In the past joint meeting at the 15th AMM between PMO and SAT, Agustin Benito Bethencourt explain ed this, which has been communicated in different occasions through the genivi-projects mailing list. Check for instance this thread.
- A first look at the data provided by Google analytics shows that the board images in the target boards wiki pages should be compressed so the pages load faster GDP-453 ANALYSIS

Users behaviours/analysis

After the GDP 11 release, with around 3-4 months of data, a deep analysis of how users travel through our wiki pages could be done. It should analysed if the theoretical critical path is working as expected or not. The following points of improvements has been identified already:

1. The jump from the target board pages to GDP in detail can be highlighted better, as stated in the image (small change).
2. GDP in Detail should include a description of each automotive component GDP-440 ANALYSIS
3. Each automotive software component should include a wiki page with a description to newbies and instructions to find out the most basic features of the component on GDP: GDP-441 ANALYSIS . This wiki page can live in the component Confluence space, but should be identified with the *gdp* label.

Services and Infrastructure

There are two sources of information that describe the conclusions below:

- A discussion over the mailing list about the needs for the GDP project in this regard
- The presentation about this topic done at the Tools Team f2f meeting during the 15th AMM. Please check the last few slides of this presentation summarizing the points described (slide 50).
- Unifying all the Tools that GDP requires under the control of the team would be a small by high impact step in the long run.

Deployment infrastructure

- Automated deployment. Currently the deployment of images is manual and requires the participation of GENIVI IT, which

stabilization or maintenance phase of the release. This has not always been the case since Master was created after GDP-ivi9 was released.
- GDP processes, tooling and management tasks: task related with managing the project, contents, events and other tasks that do not fit in any other category: GDP-11 IN PROGRESS
- GDP-ivi9 is still under maintenance so the GDP Delivery Team still provide support on it: GDP-8 IN PROGRESS

The labels used by the GDP team are listed in the label list. Labels used by other groups or people has partially been tracked also in this list.

Source mirroring

Within the Tools Team, it was decided to mirror GENIVI sources from Github internally TOOL-79 DONE . An internal Gitlab was put in place for this purpose. This services is centralised in GENIVI data center. Next steps are proposed at the Next Actions section.

(*please check below for more content*)

slows down the process and introduces risks.
    - Towards continuous deployment: the idea is to be able to remove IT from the equation first, and be able to deploy automatically as second step. This task can hosts the actions in this regard: GDP-10 TO DO .
- p2p capabilities and mirroring.
    - Including p2p capabilities in our deployment infrastructure will allow us to keep increasing the impact of our releases keeping under control the bandwidth required to assume a high demand on specific dates.
    - Including mirrors for our images will work on the same direction. We only need to guarantee the integrity of the deployed software and that we can find out the number of downloads from those mirrors in the same way we find out those from ours.
- Public/private instance to control what is not ready to distribute.
    - Not all images are meant to be consumed. There is a need to reduce the exposure of those images that the GDP delivery Team consider are not meant to be consumed for technical, legal or business reasons.
- Analytic to comprehend binaries consumption.
    - Now that we have Google analytics in place, there are the following steps to take:
        - Group those pages related with GDP that should be analysed.
        - Create goals.
        - Analyse the critical path journey, bounce rates, permanence time, recurrent visitors...
        - If we publish the conclusions of every date we track, we explain how we do it and what for, consumers will be open to us doing it since we need to learn in order to improve.
    - Track downloads
        - Find out the number of downloads of the images, when they are performed and from where keeping anonymity (no IP).
        - Publish.
- Steps forward towards reproducibility.
    - Introduce in the building system capabilities to reproduce the builds in the future like signatures, keeping the build system/tools versions linked to the images (outcomes), etc.

Sources mirroring

Mirror GENIVI sources (from Github and upstream) in GENIVI's infrastructure because:

- Increase reliability at building time.
    - This measure would reduce risks during release time (Gold Master declaration).
- Compliance (confirmed by LRT at 15th AMM in their talk): GDP-50 TO DO
    - Revision at commit stage to reduce overhead at validation stage.
    - Easier to control the software that needs to be analysed.
- Increase build performance.
    - This measure would have a positive impact on daily basis.
- Step forward towards reproducibility.
    - GENIVI would be able to have a better control over the shipped software. This might be valuable also

for Members, specially with software meant to be in production.
- Analytic to understand how source are consumed.
    - Who is using GENIVI components?
    - Which sources are consumed through Master?
    - All these answers are key to take decisions.

The next step should be to:

1. Include the generic software components used in GDP in the mirror.
2. Ensure that the mechanism of mirroring GitHub repos in Gitlab works for every change merged, at least.
    a. In a second step, for every PR.
3. Set up something like Trove to keep repos up to date.
4. Set up GDP to use Gitlab repositories instead of upstream repositories.

This topic has partially been covered in the Tools Team:

TOOL-9 DONE

Build system-infrastructure

- Capacity increase.
    - GENIVI is deploying today around 10-15 Gb of software. It is creating builds for 8 targets on regular basis. Every second reduction in building time has a significant impact over a year.
- Differentiation between production quality builds and other builds.
    - Users/consumers, contributors and Delivery team are interested in different builds. It is necessary a differentiation on which builds are targeting who.
- ACL.
    - The delivery process management tool needs to consider that different roles assume different responsibility in the process. In order to guarantee that those responsibilities can be assumed, keeping the maximum level of transparency, ACL capabilities needs to be included in the delivery management tool (currently go.cd).
- Trigger sources from GENIVI mirror.
    - In order to keep source mirrors up to date, an automated mechanism to trigger and create pull requests before merging upstream changes into our repositories will be needed. The GDP delivery team recommends to use Trove, which has been proved to serve this purpose.
- Parallelization.
    - An increase in the build capacity will allow to parallelise the build creation. This will allow the delivery team to create a more robust pre-integration stage where changes can be tested in isolation, reducing the complexity of the analysis and reporting of issues back to developers. This will have a significant impact on the desired reduction of the feedback loop between GDP Delivery Team and GENIVI/upstream developers.
- Developer build systems.
    - The wiki recommends the use of Ubuntu 14.04 for building GDP. As this release is approaching the end of its life we would suggest that the developers move to Ubuntu 16.04 - GDP-408 TO DO . Initial testing with this release has not revealed any problems.

Testing

- First stage: focus on acceptance testing. Acceptance testing would allow the delivery team to promote collaboration around testing. In order to do so, the following steps need to be taken:
    - Focus on processes and tests first, initially through scripting. This is the opposite to the current trend, that focus on automation at first.
    - In order to succeed in attracting contributions around testing, test needs to follow the same principles that coders code patches. They need to be simple, auto contained and well documented so they can be easily reviewed.
    - Complex test cases should be a collection of simple tests that should be reused every time, as a general approach.
    - Complex test cases should be also self documented, including the list and sequence of tests to be executed and why. In this regard, this is nothing but a "*declarative*" approach to testing.
    - One of the most effective types of acceptance tests for platforms/distributions is boot testing.
- Second stage: focus on boot testing automation
    - Though kernelci.org for simple systems. These systems are part of the pre-integration stage were new features and changes are isolated and tested to simplify the detection of the most common failures, simplifying also the identification of the developer responsible for the fix and the time spent to report him about it.
    - Through openQA for full systems. openQA is the most advanced FOSS tool for testing (specially installation and boot testing) complex systems. It would be a key improvement for GDP to have openQA as the default testing tool.
- Third stage: focus on production quality
    - Once a simple approach to testing and openQA/kernelci are in place, it is time to automate the testing activities integrating them in the delivery chain. In order to do this, changes in the frameworks will be needed, specially in openQA, based on perl.
- Compliance include specs, code and tests.
    - Once the GDP delivery team is able to test consistently every build and, if possible, every isolated change in a simplified staging area, it will be the time to ensure every new feature at least comes to Master with a test. The compliance program should also include test as outcome.

Other improvements

- Increase download bandwidth to improve integrator's efficiency. This action would have an inmediate positive impact on the daily work of the new GDP Team.
- Create/use a pastebin service for developers.
- Create/use a pad (collaborative real-time editor)service for notes in meetings and technical sessions.

GDP promotion and training

Hands on Sessions

As part of the AMM, a couple of Hands on Sessions are delivered using different boards. The contents related with the ones done in 2016 are published on the wiki, including the slides and exercises participants did. There are other Hands on Sessions with different goals delivered by different companies. The goal is that all of the

GDP promotion and training

Hands on Session

- In our opinion, the Hands on Sessions has a very low impact compared to the effort put by GENIVI and the team on them.
- The most interesting GDP training action would be to create a webinar and provide support to it through video chat.

trainers provide a single virtual Machine to participants including all the tools and systems used during the sessions. The goal was never reached in 2016 although significant steps towards it were taken, using the SDE.

Check for instance the tasks tracked for the:

- 14th AMM Hands on Session: GDP-25 DONE
- 15th AMM Hands on Sessions: GDP-274 DONE

Events

The GDP Delivery Team have participated in several events, promoting GDP. The wiki page GDP Out There has all the information about this promotion activities. The participations on the GENIVI 14th and 15th AMM were sponsored by GENIVI. The rest has been sponsored by Codethink in the case of Agustin Benito Bethencourt and LGE in the case of Changhyeok Bae.

The main events has been:

- Embedded Linux Conference - ELC SanDiego, US
- GENIVI 14th AMM. Paris, FR.
- Automotive Linux Summit. Tokyo, JP..
- OpenExpoES. Madrid. ES.
- Embedded Linux Conference EU. Berlin, GE
- GENIVI 15th AMM. SFO, US.

Demos

The talks during the last two conferences of the list included demos showing GDP 11 RC2 and RC3 in an RPi3 with the RPi touchscreen. The demo consisted on:

- Showing the HMI
- Through a serial cable, showing the boot process.
- Show the processes that reflect the existence of the generic components.
- Show some of the automotive components.
- Show the GDP build environment, configuration, scripts etc
- Demo how to modify the GDP, bitbake hands-on

Other promotion activities:

The GDP Delivery Team actively participates in the promotion of each pre-release and final release. The activities are listed in each release task, for instance:

- Release announcement:
  - Create the release announcement: GDP-387 DONE .
    - Lately GENIVI Marketing is participating actively in this task.
  - Publish the release announcement: GDP-388 DONE
    - This is done by the GDP delivery team through the blog section of the wiki and the mailing list. GENIVI marketing takes care of the press and media.
  - Promote the release announcement:
    - Social Media campaign creation: GDP-307 DONE . This has been done traditionally by the GDP Delivery
- Coordinate all the trainers so a single image is provided to participants, valid for every Hands on Session would simplify the set up, reducing the time spent by participants in getting the tools and reducing the required space in their laptops.
  - The Genivi SDE image is a step towards this goal and was used somewhat successfully at the 15th AMM in Burlingame. The issues with the image used in the current state at that point in time have all been reported in JIRA.
  - Having that image ready to be downloaded days before the training and notify participants would reduce the need for usb sticks and configuration issues that some participants have since they do not know in advance what the image looks like (which VM is it, how much space is needed, etc.).
- Consolidating the sessions to be experience dependant, not board/target dependant would also be beneficial.
  - The difference between the target boards are very minimal (in terms of what can be shown in 3 hours, whilst giving a general overview. GDP should continue to be generic across all targets), so having two sessions, one for beginners & one more intermediate should be considered going forward.
  - This would also make it more obvious to people attending the AMM if the sessions are suited to them, not just a generic overview of gdp/yocto/git etc.
- Given that now it is easy to consume GDP in binaries, the training sessions should be focused on:
  - Building GDP and customising it at build time. This approach requires a full training day or a webminar.
  - Work with GENIVI automotive components in f2f training sessions provided by the component maintainers. In other words, focus the training sessions in what makes GDP unique, not in what makes it a comodity.

Events

- It would be very interesting for the project to keep participating in ELC and ELCE at least with a talk, so the effort and costs contributed by Codethink has continuity.
- In order to increase the impact among system developers, it would be an interesting to have a booth at those events, together with ALS, to show GDP.
  - GENIVI Members could join the effort to reduce costs and share several interesting demos based on GDP.

Demos

- Once we have shown that GDP is based on generic GNU/Linux software components, the next step would be to focus on the automotive components that are part of the system, developed by GENIVI.
- Create a generic demo for CES and AMMs, together with potential booths at ELC (E) and ALS would be an interesting investment for the future.
  - Videos to promote through internet would also be a good marketing action.

Other promotion activities

- Promotion around the releases needs to increase in order to reach a wider audience. Each one of the last release articles has 100 impacts, which is far from enough.
  - Media articles should not substitute, but route

Team except for the last release, where GENIVI marketing has taken fully control.
- Publishing: of the social media campaign: GDP-345 DONE . GENIVI marketing controls social media accounts. The GDP Delivery Team supports the effort from Lin da Braun.

Another important activity, shared with the GENIVI community, is providing updates through blog posts of what is going on on the project:

- GDP has a blog where all the articles are published. Using the Confluence capabilities those articles are also published in the GENIVI blog.

License compliance

- GDP follows the product licence compliance as recommended by Yocto directly, and publishes this set of metadata along side every release, for every target.
- The creation of this metadata is handled by the genivi go.cd server, with the output tested / verified and passed to genivi IT for hosting. This pipeline templates can be seen here wit h appropriate credentials.
- The GDP team worked continuously with LRT to uphold licence compliance. This included manual source checking and investigational work with bitbake, including assessing the current state of spdx/fossology support within the tool. M etadata hosted by Genivi for each release has been made available to LRT, along with specific requests.
- The work in this are can be analysed in the following tickets: GDP-41 DONE , GDP-95 DONE and GDP-90 DONE , for instance.

Automotive components

FSA

“Fuel Stop Advisor” is a code to demonstrate main APIs standardized by the GENIVI Alliance under the Navigation scope. The GDP Maintainers worked together with PSA to consolidate & produce a digestible patch series to integrate into GDP. This included face-to-face meetings at the 14th AMM GDP-15 DONE

Wiki page here. Repositories: Recipe

SOTA CLIENT

Software Over The Air Client is written in the rust programming language & uses the cargo package system, provided by meta-rust. GDP-145 DONE .

The integrated example has a wiki page on the GDP wiki here. Repo sitories: Recipe & meta-rust

Lifecycle

During the last couple of 2015 and beginning of 2016, the GDP Delivery Team worked on the integration of Lifecycle into GDP, which required also development work. This effort was de-prioritised

potential users and contributors to the release articles, that include the relevant links to further information like the Download page or the build instructions, for instance.
- We recommend to include G+ as a channel of our social media campaign. It is the default social network for system developers.
    - For potential new users, creating a GENIVI group or increasing the current promotion done by the GDP Delivery Team of GDP in automotive and embedded Linkedin groups can help. That group can be created together with AGL, as the IRC channel to have an initial critical mass.
- The GDP 11 final release media coverage should be tracked in order to compare it with the previous release, but above all, with future releases.
    - Add social media impact and hits on relevant wiki pages and communicate them to GENIVI marketing.. GDP-139 DONE

License compliance

- During the 15th AMM, Claus-Peter Wiedemann and his colleague presented to Members only the conclusions of the evaluation of GDP-ivi9 licenses.
    - The GDP Team will need to work together to address those conclusions applied over generic software components, since Yocto is not providing us today a 100% complete overview of the licenses involved in GDP.
- Currently there is a directive from the Board to not use GPLv3 code in Master/GDP. There is a need to define how this statement affects GDP and the engineer.
    - Check the Challenges section for more insight about what we recommend in this regard.
- GDP developed coded licenses are checked regularly, before publishing. Today, generic software component licenses used in the platform are checked during the validation stage, relaying on the information provided by Yocto.
    - In order to speed up this process a proposal to mirror sources and define a mechanism to highlight potential conflicts before the integration stage should be put in place. Otherwise, the validation process (check licenses beyond what Yocto provides) can take too long to be effective, or should be done after the release, like it did happen for GDP-ivi9.
- The collaboration with LRT team is key in the future to address these challenges: GDP-123 TO DO

Automotive components

Lifecycle

- As reported, in order to integrate Lifecycle into GDP, development effort is required. Since development is an activity that has a huge impact on the GDP Delivery Team, our recommendation is to assign this task to a different team/individual. The tasks to be done are described as subtasks in GDP-7 ANALYSIS .

in order to focus on the update and stabilization of the base system in order to create GDP-ivi9 on time for the 14th AMM. The conclusion of the work was described in the following task:
GDP-7 ANALYSIS

Generic components

Weston, ivi-extension, ivi shell & qtwayland

- Each baseline upgrade to GDP (meta-ivi) brings in a new version of Weston (reference Wayland compositor) into the system. Weston is available in the core poky layer, however the genivi baseline is often ahead of this
  GDP-51 - Update weston in GDP-ivi9 to version 1.9 DONE
  .
- New versions of Weston also brings in a new version of ivi-extension (Genivi Layermanagment API) which the GDP HMI needs to be configured for GDP-163 DONE .
- Upgrading Qt (inline with poky release version in use for meta-qt5, currently 5.6) leads to ivi-shell support in qtwayland work. ivi-shell is not in upstream Qt till 5.7 (Issues here, GPLv3 etc) so the substantial series to qtwayland needed to be substantially reworked each upgrade GDP-229 DONE .
- Weston support for Raspi was also a big item of work, adapting to drm-backend & following the upstream progression of the vc4 driver into the official raspi kernel provided success, including 'Day One' supports for the official raspi touchscreen, integrating DSI driver support not available in the yocto BSP GDP-206 DONE

Audio Manager

GDP Delivery team worked on bringing Audio Manager 7.4 to GDP. Audio plugins held in meta-genivi-dev (for pulseaudio) are unmaintained and, due to the changes that AM 7.4 introduces compared to AM7.0, they do not work. So we decided during the GDP 11 release cycle to stay with AM 7.0 and propose at the 15th AMM, after GDP 11 release, to drop them. Since the GDP 11 release schedule was changed, the discussion can be postponed.
GDP-336 DONE . Up until AM 7.4 the patches have been rebased for each AM upgrade going from meta-ivi7 to 11 in terms of GDP upgrades.

Meetings and reports

The GDP Delivery Team participates in the following meetings:

- GDP weekly call: driven by the GDP Delivery Team
- PMO call: participation
- BIT call: participation
- Tools Team: not mandatory but this is the group where key tools for GDP project are discussed and recommended to the OSS team, which decides priorities and workloads for GENIVI IT.
- GDP maintainers meeting: weekly meeting where maintainers sync. Done through Google hangout.

FSA

- FSA should be integrated with the new Application Launcher in order to be launched: GDP-447 TO DO .
  We think this would be a good feature for GDP 11.

Audio plugins

- Audio plugins are unmaintained. This fact has prevented GDP from moving away from Audio Manager 7.0 towards 7.4, as reported below in the Audio Manager section.. The GDP Delivery Team recommended to open a debate after GDP 11 release in order to decide what to do with this.
- As a general rule, we support the idea of dropping all unmaintained code. This case should be no exception.

RVI CORE

- The Remote Vehicle Interaction (RVI) Core Node Server is provided by meta-rvi and depends on meta-erlang.
  GDP-183 ANALYSIS . Having a demo at CES showing the capabilities of RVI using GDP would be a major hit.
- Repositories: meta-rvi & meta-erlang

Navit

- Users are reporting problems with Navit.
  GDP-324 ANALYSIS . As is mentioned in the ticket, Navit is in general unmaintained.
- We recommend to drop it since it affects the reputation of the entire system and does not add value.

Generic components

Audio Manager

- In order to update Audio Manager from version 7.0 to version 7.4 GDP needs to get rid of the audio plugins, since they are unmaintained. GDP-336 DONE
- Define a sustainable solution for audio in GDP. The plugin system increases the maintenance effort required.
  - Evaluate the proposal from François Thibault Director of Innovation Audiokinetic Inc.to include as an option their audio engine, following a similar approach GDP currently follows with BSPs. The proposal is currently on the PMO plate.
  - It is not ideal since involves proprietary software. Is there any other viable solution?

Qt

It is not possible to create today a system that excites developers without GPLv3 code. Qt is the number 1 technology of its kind and the main community we can rely on to develop on top of Master and GDP. The Board decision to block GPLv3 code in GDP Master needs to be revisited for three main reasons:

- Staying in Qt5.6 will require an engineer effort that will grow overtime, consuming the already limited resources.

Weekly report

GDP team published every week a report about the activity done. The report has two parts:

1. The first one is created manually by the Project Manager every week on the wiki. Check an example.
2. The first part is exported to .pdf. The second part is a .pdf document exported out of the GDP Delivery weekly report. Both documents are merged into a single .pdf document that is attached to the GDP management wiki page.

Once the wiki and the full .pdf version has been attached, a notification to the genivi-projects mailing list is sent.

GDP call every week

- GDP Delivery team is in charge or organising a weekly call through Webex. The team uses a Windows laptop and the Android application.
- An invitation to join the call is sent to genivi-projects mailing list the day before, advising the agenda.
- Before the event, a second mail is sent including the weekly report. Usually the meeting starts by going over the report.

Other reports

1. At every AMM, two reports are delivered. The first one is a slide deck for an 10 min overview as part of the Expert Groups review session. A second slide deck is prepared as part of a shared session with the Baselines coordinator during the open days of the AMM. The slide deck supports a presentation of about 20 minutes. Check those slide decks in GDP at AMM wiki page (only the public one).
2. As result of releasing GDP, JIRA creates a list of completed tasks for each release that is being published in the GDP Release report: tasks completed wiki page.
3. GDP Delivery Team created a Kanban board to visually follow the evolution of the tasks from the team. You need to be logged in to see it. We haven't been able to make it public, although we have tried.
4. Stats about tickets works well to understand who does what. This report is provided by JIRA.
5. Check the section Relevant Links of the Weekly Report with further stats.
6. The GDP lead participates at the PMO call every week. When required, she provides there a report of specific topics. It also serves as escalation point.

- Our target groups requirements will evolve, needed higher capabilities that only newer versions of Qt can provide. Master-GDP will be at a disadvantage.
- The whole story about bringing innovation to the platform cracks at a very important point. Today's story behind the platform becomes inconsistent.

Our recommendation is to move to Qt 5.7 GDP-220 DONE, limiting the restriction of using GPLv3 code to GENIVI developed components. Check more about this topic in the Challenges section below.

Meetings and reports

Meetings

- In order to increase the audience of the GDP Weekly Call, it would be interesting to evaluate the shifting of the time of the call to allow far east and US west coast participants to attend at work times.
- Since Changhyeok Bae Bae will continue to work as maintainer, it will be needed to define a meeting to sync with him if there are not timezones overlap with his work time (as maintainer) and the new GDP Team.
- Having weekly or bi-weekly meetings through IRC is a good way of increasing the audience of the weekly call. We recommend to try it monthly at first.

Latest reports

- GDP retrospective: 2015Q4 and 2016 (GENIVI Development Platform GDP)
  gdp report
- GENIVI Development Platform management (GENIVI Development Platform GDP)
  gdp gdp_maintenance management jira report rvi sota
- GENIVI Development Platform management > GDP_report_09-11-2016_week_44.pdf (GENIVI Development Platform GDP)
  gdp delivery report
- GDP Delivery Report Week 44 2016 (GENIVI Development Platform GDP)
  gdp report delivery
- GDP Delivery Report Week 43 2016 (GENIVI Development Platform GDP)
  gdp report delivery

Transition

The following actions would be needed to transfer the control of the project over the new team:

- Repositories admin/write rights at Github
  - Remove write rights to the current team.

        - Remove admin rights to Agustin Benito Bethencourt
        - Change maintainers group members and update the MAINTAINERS file.
- Admin rights at JIRA and Confluence
    - Remove Agustin Benito Bethencourt admin rights.
- Take over the GDP weekly call on Nov 16th 2016
    - New PjM send the invitation the day before
    - Ask Jeremiah Foster to provide rights to create/manage the meeting at Webex.
    - Create the report for that meeting.
- GDP maintainers weekly meeting
    - Create/manage the sync meeting with Changhyeok Bae.
- Join the #automotive channel in IRC.
- Assign all the current tasks from the GDP Delivery Team to the new PjM. All task should be currently assigned to the former PjM, Agustin Benito Bethencourt
    - Explore to assign them automatically.
- Where to start?
    - The recommendation is to start reading this report, then read the wiki (GDP Home page), then check the repositories (1)(2) and finally go over the tickets (GDP Kanban board).

Challenges and Recommendations

GDP project challenges

GDP Team would like to describe briefly the challenges the project is facing today or will need to face in the coming months in order to grow significantly, specially in the number of contributors:

1.- Culture

Having the right balance between a code first and a spec/compliance approach will determine the future of GDP to a great extent. Today it is unbalanced. There is no successful Open Source ecosystem based on compliance programs as a necessary step to integrate code that brings the attention of external contributors. Code first approach is a key prerequisite to be successful in creating a vibrant R&D ecosystem. The GDP Team is a key piece to consolidate the code first approach within GENIVI:

- Act by example, following Open Source best practices with a didactic approach.
- Support GENIVI developers in following the contribution policies, based on Open Source standards
- Promote that developers bring software to Master directly, applying the "software first policy", as initial step to get into the GENIVI ecosystem. This is especially true for small companies.
- Be vocal when basic processes are challenged by Members or GENIVI management. A significant part of the trust the community has on the GDP team comes from the visualization of its members as Open Source advocates within GENIVI.

The growth in the number of contributors and the amount of software that lands to GDP will be a direct consequence of the above points.

2.- License policy for GDP/Master.

The success of GDP will depend on how the license policy (No GPLv3) established by the GENIVI Alliance Board is applied.

- GPLv3 code is already part of GDP/Master. There is no sustainable way to create a GPLv3 platform and maintain it with the current level of investment.
- The policy is incompatible with increasing our impact in the application/UI space. Qt is the only valid alternative as ecosystem nowadays. They are GPLv3 based.

Codethink believes this policy applied to an Open Source R&D organization that wants to develop a platform based on available generic components is unsustainable for Master/GDP. It will represent a significant challenge in the short term to the GDP team. The effort to stay in compliance with it will increase over time.

3.- Relation between development and delivery

GENIVI still have a significant work to do to bring development closer to delivery, which would shorten the space between developers and consumers/contributors:

- The public mailing list reduces this distance. The GDP team should keep the lead in the mailing list and keep promoting GENIVI

developers to be active on it, supporting the Community Manager efforts in this regard.

- A significant part of that relation between development and delivery within GENIVI goes through meta-ivi, which is acting as a funnel. Separating compliance from integration and restructuring meta-ivi in functional layers, merging the release manager into the GDP Delivery Team would help to close the current gap.
    - Since a few months ago, the GDP Delivery Team and the meta-ivi maintainer are working with the same software versions so the collaboration has been possible and fruitful. If the current meta-ivi setup remains, increasing-improving this relation will only bring benefits to both, meta-ivi and Master/GDP.
- Bringing the compliance process to the open, keeping the decisions around it restricted to Members, would improve the current understanding of how GENIVI operates, with a positive impact in reducing the gap between delivery and development within GENIVI. It would also ease the collaboration between meta-ivi maintainer and The GDP Delivery Team.

4.- Target group for GDP project

Currently Master targets automotive system developers and GDP application developers. The success of an Open Source platform project like Master/GDP, depends to a great extend on the size and quality of the contributors base contributors. So contributors, not consumers, should be the target at this stage of the project, that is automotive system developers. Professional application developers will come to the GDP project if:

1. There is an expectation of a market for their application, which which requires that OEMs and Tier 1s (potential customers) adopt the platform in production in the near future.
2. The platform is solid and robust today so application developers do not need to spend time in platform related activities.

To address the first point we recommend to create a new outcome, designed to be consumed by OEM and Tier 1s in production. GDP is not designed for that. To address the second point we recommend to increase the overall manpower/effort on the platform itself, investing in tools and infrastructure that allow maintainers to do more with less. In this last point, the action taken by GENIVI to consolidate the infrastructure and services will have a very positive impact. More actions in this regard will need to follow.

5.-Management overhead

Since GDP has been reset and the project is developed within a consortium, an additional management effort during these few months was expected. Codethink addressed this need. The effort has increased over time getting to a point in which an analysis is needed in order to keep the management effort required under control:

- The amount of management effort per engineering hour is currently high...
    - The number of people the team lead (manager) needs coordination with within GENIVI is high: PMO, Architect, Community manager, Marketing, CEO, BIT coordinator, Tools Team....
- ... and growing.
    - The engagement with companies and contributors is growing. As the community grows, the management effort grows accordingly.
    - In order to improve the current feedback loop with GENIVI developers, a more intensive relation between GDP and Expert Groups will be needed, increasing the management effort.

Some of the variables that affect the amount of management effort required from the GENIVI Delivery Team are under GENIVI control. We recommend the following:

- Work on standardizing processes, specially related with the release.
- Consolidate release schedules: timelines, merging windows, etc way in advance.
- Route management related conversations through the existing channels: genivi-project mailing list, IRC channel and GDP call.
    - The goal is to reduce the number of calls /time required to coordinate the project.
    - Bring maintainers to the public mailing list so interaction between integrators and developers does not require management effort.

This point does not apply to technical management which will grow as the number of contributions do. This area is not challenged at this stage.

6.- Scalability

Related with the previous two points, in order for GDP to scale up, it is needed to:

- Increase the investment on infrastructure and services around the project. Today GDP is producing around 15 Gb of information per release. This is in the order of software present in a car. The current infrastructure and services are insufficient to increase the quality of the outcome or the amount of software released.
- Reduce the feedback loop between developers and integrators.
- Reinforce the Code first path within GENIVI.
- Consolidate the Open Source project practices, specially when dealing with potential first time contributors, welcoming new code and working to consolidate their commitment to GDP by assuming responsibility in the project.

7.- Collaboration with AGL

- AGL is getting a significant momentum, especially among smaller companies, based on its Code First approach. Today its target group and base platform is similar to GDP. The GDP Team and AGL development team collaboration is working but there is room for improvements.
- A re-structure of meta-ivi into a set of functional layers including a common base layer with AGL, would also heavily improve the current situation.
- Looking for a complementary, instead of competitive, approach to the GDP platform in relation with AGL would also improve the collaboration among both projects. This can be done by targeting automotive system developers at this point, instead of application developers.

The second point requires agreements at the management level while the third one at an executive level. They would have a positive impact on the technical and collaboration side, so on the daily work the GDP Delivery team has to face.

The above ideas has been described in one way or the other in the following documents in different mails in the GENIVI public mailing list or open calls:

- Talk at 15th AMM (public session): GENIVI Software as Portfolio. Where can we go next & how?
- Talk at the 14th AMM (Members only session): One organization - one delivery chain for a portfolio.*Note: you need to be a GENIVI member to access these slides.*
- GDP viewed as portfolio document, sent in June 2016 to the public mailing list as attachment.

People and links

GDP delivery Team

The GDP delivery Team was formed by:

- Tom Pollard, engineer at Codethink Ltd
- Robert Marshall, engineer at Codethink Ltd
- Jonathan Maw, engineer at Codethink Ltd
- Agustin Benito Bethencourt, coordination at Codethink Ltd
- Changhyeok Bae, engineer from the community.

Communication channels, links and resources

The project communication channels are:

- GDP project mailing list: public mailing list. Central discussion point.
- GDP IRC channel: #automotive in irc.freenode.net, shared with AGL.
    - Check the channel log

The key links are the following:

- GDP project wiki
- GDP management tool
    - Tasks
    - Policies
- GENIVI at Github
- Blogs
    - GDP blog
    - GENIVI blog

Do you want to contribute?

- How to contribute to GDP
- GDP repositories in GitHub:
    - genivi-dev-platform
    - meta-genivi-dev
    - gdp-test-suite
- Report bugs: bug tracker