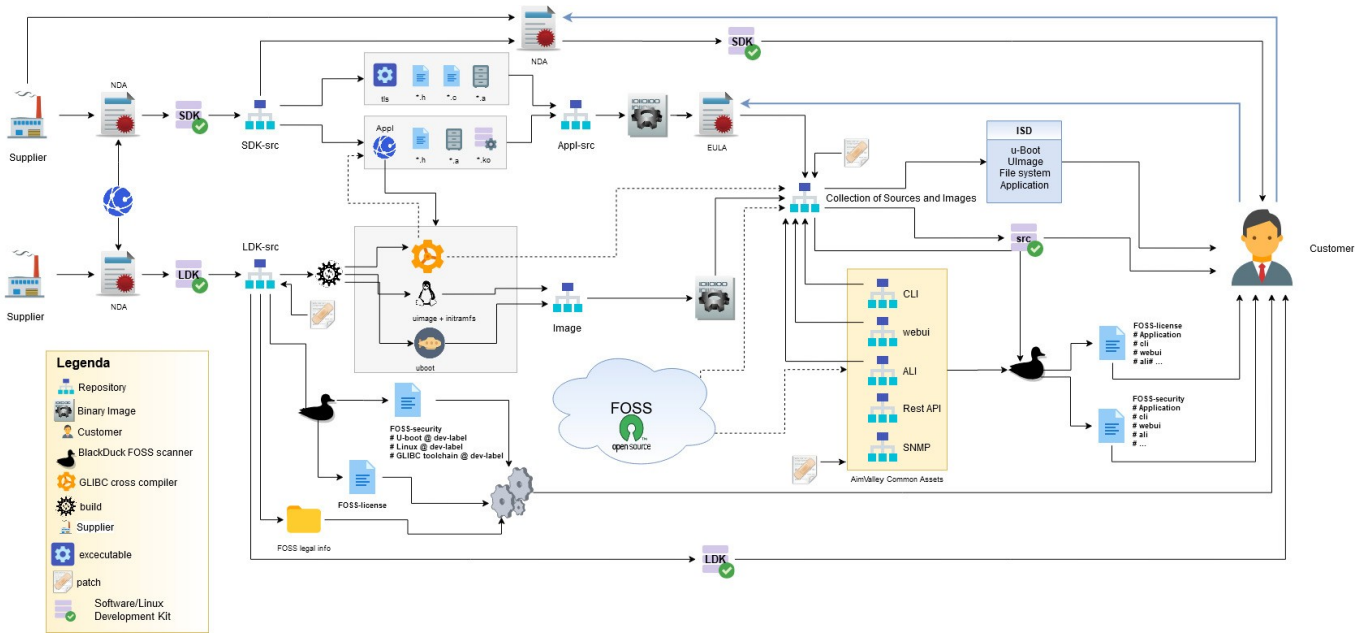
Free & Open Source Software

White Paper

AimValley

AimValley

Free & OpenSource Software White Paper



Embedded Development with Free and Open Source Software (FOSS)

Introduction

AimValley is a world class engineering and innovation center that designs and builds the best solutions for a connected world. We bring over 15 years of experience in telecommunications and complex system development, electronics design, ASIC/FPGA and embedded software development.

Free and Open Source Software (FOSS) has become critical for almost every organization.
Its growth in the past decades has been exponential. Today it is used by 75% of the organizations using software code. The process of mass collaboration and peer production sustains innovations in mature software markets. The application of FOSS extends the useful life of any software asset.

Benefits of Open Source Software

- **Security:** thousands of community members referring, contributing and testing a piece of code,
- **High Quality Software:** created by developers with the end-user in mind (themselves),
- **Cost efficient:** significant reduction of development time through re-use of code,
- **Portable:** no Vendor lock-in or inability to customize,
- **Accessible:** source code is available,
- **User friendly:** ability to install several times and use from any location,
- **Worldwide support:** freely available and easily accessed online,
- **Flexible:** scaling & consolidating, various options for clustering, load balancing.

AimValley

Utrechtseweg 38
1213 TV Hilversum
The Netherlands

phone +31 35 689 1900
sales@aimvalley.com
www.aimvalley.com

AimValley

Free & OpenSource Software White Paper

Free and Open Source Software and the Embedded World

AimValley provides connectivity solutions most of which are embedded solutions. Embedded solutions often come with a lifetime of more than 10 years and with limited maintenance windows and opportunities for upgrades and applying patches. As a result the software needs to be robust, secure and preferably first-time-right. For the Healthcare, Aerospace and other demanding markets, standards exist that require even more proof of fitness. This in contradiction to 'non embedded solutions' or applications where maintenance and upgrade windows are more frequent.

Can we still use FOSS in embedded applications? The AimValley answer is - yes we can!
Depending on the application and requirements, there are some precautions that need to be taken to guarantee that the end product meets your expectations.

At AimValley we have developed our own Quality Assurance Process for developing any product with or without Open Source Software, while addressing the possible risks.

Quality Assurance Process

Security is a hot topic in today's modern software development methodologies. Security mitigation starts early on during development and continues during maintenance and support, after many years of product release. The AimValley Quality Assurance Process provides an approach to mitigate any security issue in each step of the development process.

AimValley uses a formal development process which is the key to deliver high-quality and robust software. This development process consists of a software version control system combined with a controlled access and change management system and tooling to enforce and facilitate review and traceability of all changes. Together with the described FOSS quality assurance process we guarantee a high quality and robust development process including the use of FOSS.

The AimValley Quality Assurance Process includes the following steps.

1. Security Engineering
2. Component Selection
3. Definition and Architecture
4. Design and Implementation
5. Integration and Verification
6. Maintenance and Support

Security Engineering

Building a secure system starts very early on in the development cycle during the systems engineering phase. During this phase the AimValley systems engineers determine the right level of security for the application together with the customer. The resulting system architecture and security architecture are captured in verifiable features and requirements which are used in the following development and verification activities.

Utrechtseweg 38
1213 TV Hilversum
The Netherlands

phone +31 35 689 1900
sales@aimvalley.com
www.aimvalley.com

AimValley

Free & OpenSource Software White Paper

In today's world security challenges transcend the individual system. From day one, the architecture is analyzed for attack surface and together with the customer, the risks and impacts are determined. The resulting Security Requirements Document (SRD) also known as the security target, provides traceable security requirements, the rationale behind them, as well as use cases and other counter-measures to achieve the level of confidence needed given the context of the project.
Maintenance aspects (e.g. secure deployment, patching strategy) form an integral part of the SRD.

Component Selection

Depending on your license requirements, we select the appropriate FOSS components, extend or modify existing FOSS or develop custom functionalities. FOSS licenses come in many flavors ranging from permissive to highly reciprocal. Based on your requirements regarding IP disclosure, the associated FOSS license and how the FOSS is used in the actual product, the FOSS components are selected or discarded.

Next to the current and outstanding vulnerabilities of an eligible FOSS component, assessment of operational risk is an important factor in the process for selecting a FOSS component. At AimValley, we assess the likelihood of possible vulnerabilities now and in the future. Therefore, AimValley also looks at the open-source project activity, the reported vulnerabilities over time, the number of lines of code, the number of participants and the release cycle of the open-source project. Projects that show little activity and a very low release cycle might be discontinued in the future and the maintenance of the FOSS component falls back completely to the owner of the product being developed. Choosing the right FOSS component can make the difference between a first-time-right product or a vulnerable product, due to continuous extra development costs and/or ultimately being discontinued due to vulnerabilities.

Definition and Architecture

Another key process step is Threat Analysis; assuring that the architecture/design is security aware and provides sufficient protection against any foreseeable threats. The threat analysis identifies the inter-faces, actors, threats, data, services, classification and mitigation. Design choices such as segregation of services in virtual containers, flash configuration and data storage allocation can provide a higher level of security. Together with our solid development process focused on defensive programming, reviews and traceability of changes, AimValley can guarantee a secure and high quality product.

Design and Implementation

During the implementation, code reviews, continuous daily builds and automated regressions, static and dynamic code analysis tools such as Coverity from Synopsys and Valgrind are used to guarantee quality and robustness. During this phase also fuzz testing is performed where necessary.

Utrechtseweg 38
1213 TV Hilversum
The Netherlands

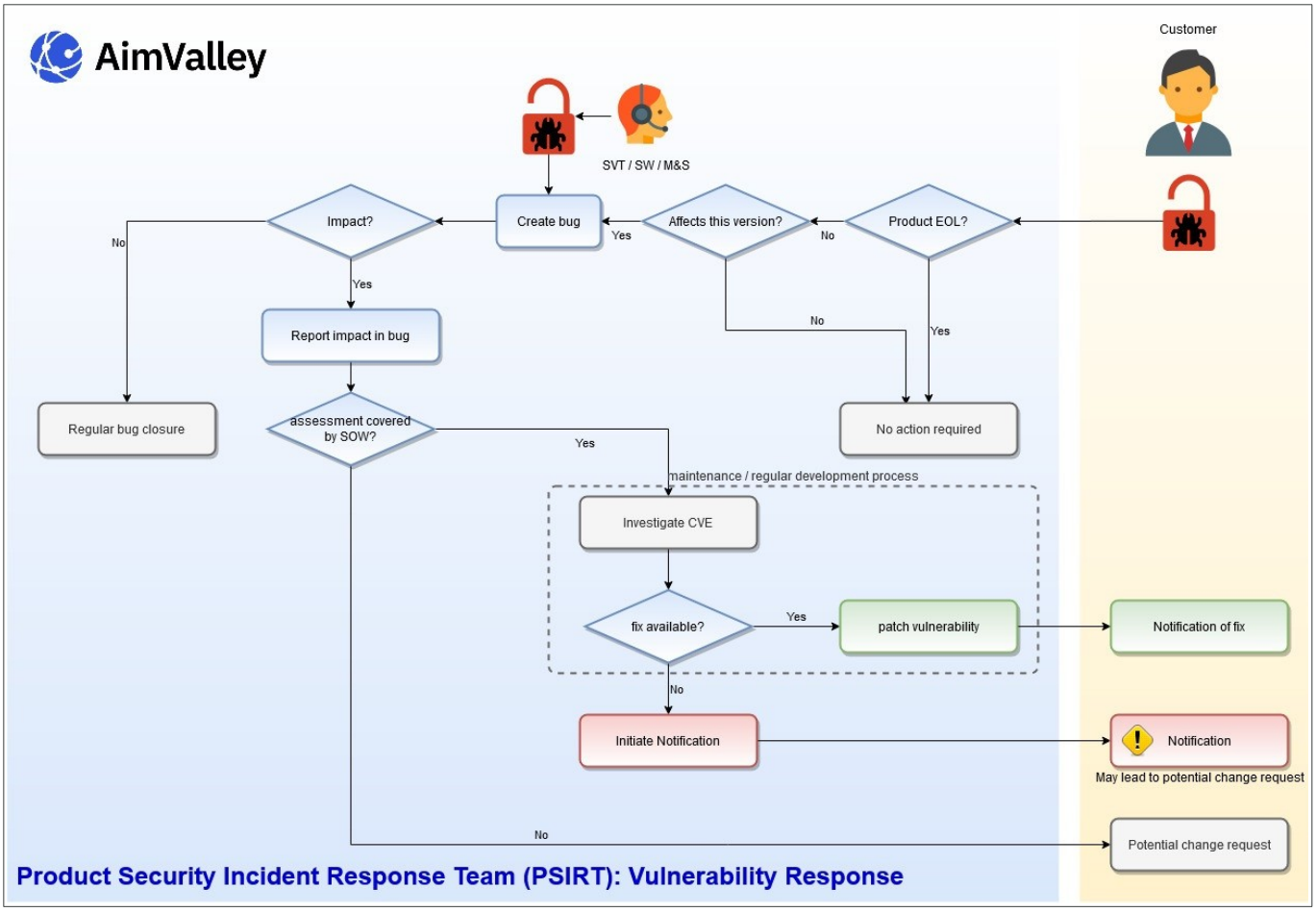
phone +31 35 689 1900
sales@aimvalley.com
www.aimvalley.com

Integration and Verification

With a strong focus on security during architecture, design and implementation, AimValley performs penetration (PEN) tests with OpenVAS, Nessus and Kali during the integration and verification phase. PEN tests are used to catch all unforeseen cases and verify if the design is really up to par. There is one additional step that can be taken here and that is to engage an ethical hacker/specialist to actually attempt to hack the system.

Maintenance and Support

After product is released and being deployed in the field, it is very likely that new vulnerabilities are detected in existing FOSS and the product is amenable to a security breach. AimValley has a dedicated team available that monitors and tracks vulnerabilities using BlackDuck and can provide patches and upgrades as a custom service.



AimValley
Utrechtseweg 38
1213 TV Hilversum
The Netherlands
phone +31 35 689 1900
sales@aimvalley.com
www.aimvalley.com

Vulnerability Detection in Open Source

The first step in the process and possibly the most important is the identification of the open source components. All sources used during the software development of a product are scanned for open source components. Using BlackDuck enables us to identify any possible component that might render an issue, concerning licensing, operations or security.

BlackDuck provides information regarding licenses, vulnerability, and operational risks. A BlackDuck scan of embedded sources provides in most cases a vast amount of data and requires manual investigation, assessment and annotation before it can be processed in the product. Note that there are ways to reduce the number of scanned sources for yocto-based BSP projects such as BlackDuck's BITBAKE detector and other alternative methods that process the recipes and layers, and thus limiting the amount of data that needs to be manually processed.

However manual investigation is still required to filter out wrongly detected vulnerabilities and identify undetected components. It is also applied to mark components/tooling that are not used in the final product.

Board Support Packages

Board Support Packages (BSPs) are provided by a hardware supplier. BSPs are always based upon a (not so recent) Kernel release and thus inadvertently pose a possible risk to security. In practice BSPs are rarely updated by the hardware supplier. Furthermore, software modules are often modified by the supplier to comply with the hardware.
Open source patches for vulnerabilities are often based on more recent kernel releases. Back porting of fixes is complicated and sometimes even impossible to implement without the risk of introducing new vulnerabilities of their own.
Next to these issues the Common Vulnerabilities and Exposures (CVEs) listed are often difficult to reproduce and it is equally hard to demonstrate that the fix was successful. In addition, many CVEs are poorly described and difficult to assess for relevance.

The situation is different with respect to Linux board support packages, as these sources are often provided by the supplier of the processor platform. These Linux development kits (LDKs) always include support for a multitude of boards and processors, not all are relevant for a particular project. This aside from the plethora of Linux services that are included. At AimValley we also scan these BSP sources and use the components list generated by the Linux development kit as a guide to select the right components to assess and annotate. It takes time to build-up your experience in an operating system like Linux.

So leave it to the professionals!

At AimValley we have over 80 experts with more than 25 years of experience in building BSPs from scratch and using BSP/LDKs from various vendors in embedded systems.
As expected, the FOSS benefits come with risks. Some are insignificant and easily mitigated, others are more complex and require a much more rigorous approach.

AimValley
Utrechtseweg 38
1213 TV Hilversum
The Netherlands
phone +31 35 689 1900
sales@aimvalley.com
www.aimvalley.com

Quality Assurance

The level of quality assurance differs per market segment. Some markets allow frequent updates and are more forgiving to issues than others. Others are very strict and leave very little room for updates and issues. The latter are often backed or enforced by standards and regulatory committees and boards before a product is accepted.

AimValley can support both worlds. Although our process is a formal one, the activities we execute per project are flexible. This means that we follow the process in all cases but the level of security mitigation, analysis, testing, verification, CVE handling and auditing is agreed upon with our customers.

For example; a customer who requires only a binary application delivery and does not need any further support. In this case we provide the customer the means to rebuild the custom developed kernel modules when they are ready to upgrade to a newer kernel version. The binary application remains the same and does not need updates or patches.

Another example is a customer who requires deliveries in source, including the ability to extend the application and BSP. For this customer we deliver the sources along with the complete development environment, documentation and training, including a host simulation and target build.

For customers who require maintenance & support, we also maintain lab-space and hardware setups in order to provide support all year around. In this case we provide updates with patches to resolve known CVEs and other issues.

Based on our robust development process and Quality Assurance, we are confident that FOSS can be used in embedded systems and that AimValley provides you with the support to safely do so.

Summary

Business Benefits

So why let AimValley develop your product with a Free and Open Source Software component?

- High Quality Software
- Reduced development time → Faster Time-to-Market
- Flexible/Scalable/Portable
- Cost Efficient

No Risk → Aimvalley Quaity Assurance Process in place.

For more information or a demo contact our sales team.

Utrechtseweg 38
1213 TV Hilversum
The Netherlands

phone +31 35 689 1900
sales@aimvalley.com
www.aimvalley.com