Reduce Design Time and Product Lifecycle Costs with Functional Blocks Common to Designs and Test Fixtures

Stephen Golemme, FMC Technologies Schilling Robotics

Abstract
This paper proposes an integrated design and test strategy which uses functional design blocks common to both designs and test fixtures, with the aim of saving time and money. Benefits of this strategy include reducing total product lifecycle cost by decreasing product and test development times, improving sustaining engineering processes, and reducing unique component counts across products and test fixtures.

Benefits and drawbacks to such a system are explored for a company that produces low volume, long life products. Benefits analysis is applied to software development and maintenance, schematic design, printed circuit board layout, hardware design, and fixture development for verification, qualification, and functional tests. A brief explanation is provided regarding how existing commercial functional test platforms can be inadequate in a low volume, long life integrated systems production environment. Analyses of process changes and technical hurdles that will need to be overcome while deploying such a system is examined. Finally, a phased approach is suggested that will minimize disruptions to daily operations; minimize any significant, up-front expenditure; and integrate into existing systems.

Glossary
ADC – analog to digital converter; BIST – built in self-test; BOM – bill of materials; CPLDs – chip programmable logic devices; DAC – digital to analog converter; ERP – enterprise resource planning; I/O – input and output; JTAG – Joint Test Action Group, refers to a standard test and programming interface implemented on many integrated circuits; PCA – printed circuit assembly; ROV – remotely operated vehicle; CAN, RS-232, USB – various serial communication standards; TTL – transistor-transistor logic, refers to TTL compatible logic levels of 0-5 volts

Context
The company designs and manufactures integrated systems that primarily serve the subsea oil and gas industry. Specifically, the company makes subsea work class remotely operated vehicles and subsea robotic arms. Product prices range from tens of thousands to several million dollars and have life cycles of 20 years or more. Systems are highly customizable and are manufactured in low volumes.

At many companies, including ours currently, designs are organized by major subassemblies and then by discipline. A major assembly structure consists of a top level work instruction, drawing, and bill of materials (BOM) consisting of mechanical and electro-mechanical components and software files. Most major assemblies include at least one PCA, which could consist of over 1,000 total components and over 100 unique components. System PCAs have defined communication interfaces, which include hardware and software that can be common to multiple PCAs. They can also have other common components, such as those found in power supplies, processors, filters, and other circuits. Currently, these functions are copied and pasted onto other PCAs or the same functions are designed again, slightly differently, on each PCA. There may be top level schematic diagrams for electronic and hydraulic systems at the system level, schematic diagrams at the PCA level, as well as other documents at various levels.

Common Functional Blocks Strategy
The common functional blocks strategy being proposed involves segmenting and tracking designs functionally rather than to how they are built. Instead of creating a PCA BOM with over 1,000 individual components, the BOM can be created with five to ten different blocks that are shared among several PCAs with similar requirements. Instead of having designs segmented into mechanical, electrical, and software sections, designs can be segmented by functions that consist of hardware, electronics, and software. The blocks consist of schematics, components, hardware, and software. The blocks would also be used to design custom functional test fixtures. In this way, functional test fixtures would have interfaces that match products and inherit the sustaining engineering benefits from the products.

A simple example of a common functional block is an RS-232 (a serial communication standard) to TTL (transistor-transistor logic) block with a defined data stream. The block contains a schematic of a serial level converter and its associated parts as well as a standard connector. The block also contains a serial data interface specification and a software block that interfaces with a common processor block. Finally, the block includes a PCA layout.

First, the makeup of a common functional blocks based system will be described. Then, the benefits of common functional blocks, including reuse, design verification, sustaining activities, BOM item count, and test endeavors will be explored.

Common Functional Blocks System Management

The common functional block based system proposed is essentially a logic based documentation scheme of part and schematic linkages based on functions. Manufacturers have a long history of documenting physical linkages of parts and assemblies and in many cases, the same principals can apply to functional linkages. This section examines functional block levels, proposes four types of blocks, and explores linkages to the physical BOM structure.

Just like physical BOM structures, functional block structures have blocks of varying complexity. There are innumerable methods of dividing and implementing logic blocks. One method is to look at four levels of electronic building blocks in electronic systems and then propose a flexible common functional block based system that maximizes value from common functional blocks.

Common Functional Block Levels

Electronic building blocks in electronic systems can be sectioned into four levels: completely custom, reuse, platform, and reconfigurable.[1] The lowest level of common functional blocks is to have everything designed completely custom, where nothing is copied from anything. This provides the most flexibility in design activities, but there are no savings over the product lifecycle. The next level of functional blocks is reuse, informally known as copy and paste. Design time is reduced by the time that would be required to re-design the blocks that are copied. However, this can lead to errors being copied into new designs and a lack of traceability regarding what was copied from where. This can also lead to sustaining activities being performed repeatedly on the same circuits on different designs by different engineers.

The next level is platform based block design. The time spent designing each block increases slightly due to the additional documentation required and the additional effort spent in making each block reusable. However, total design time decreases, especially as more and more blocks become available. A big benefit is that the common functional blocks themselves are testable, so the product can be designed and tested in smaller pieces. Another big benefit occurs across the remainder of the product lifecycle, as sustaining engineering time decreases substantially. Issues such as transient problems and obsolescence can be troubleshot at the block level and fixes can be rolled out across all designs. Test fixtures can use the same blocks as the designs and can be quickly designed to interface with the common collection of blocks. It becomes easy to see a collection of designs that have the same few blocks reused in them, which can lead to consolidation of assemblies and test fixtures that share similar types of blocks. The efficiency of sustaining activities is increased by focusing on collections of PCAs with common functional blocks instead of single PCAs.

The level of commonality above platform blocks is reconfigurable blocks. These designs involve interfaces that are reconfigurable on their own, usually with software. They can consist of CPLDs (chip programmable logic devices) to a system on a chip with high bandwidth, high range ADCs (analog to digital converters), and DACs (digital to analog converters). Reconfigurable blocks can extend into systems that can mechanically reconfigure themselves. Reconfigurable electronic blocks would have all of the sustaining activities and test fixture benefits of platform blocks, as well as the additional benefit of hardware that can be reconfigured for different applications. However, design costs would be substantially increased and product hardware costs would also be increased to support such a generic system. Electrical characteristics of signals (particularly high speed signals) with current hardware options cannot economically support a completely software defined system. Mechanically reconfigurable interfaces are cost prohibitive with available production technologies for most applications.

Despite the initial increased costs, a platform based block system that includes reconfigurable software defined elements (where practical) can result in the lowest product lifecycle costs for low to medium volume electronic product manufacturers. Platform blocks reduce design and sustaining engineering time and provide traceability. Common functional blocks combined with CPLDs and other flexible software systems allow the same hardware to be used in multiple applications, increasing the production volumes and decreasing BOM part count. The traceability a platform based block level affords over the more common reuse block level provide significant payoff over the life of a product. The payoff increases as the product life extends.

Proposed Common Functional Block System

This paper proposes a completely functional BOM structure in addition to the physical BOM structure that is composed of four types of functional blocks: elementary blocks, basic blocks, composite blocks, and mission blocks. This functional BOM structure would exist in addition to the physical, manufacturing BOM structure. Elementary blocks are the simplest block, designed with a single tool to accomplish part of an action, and basic blocks are designed to perform a specific action. Composite blocks perform a task and consist of a mix of elementary, basic, or other composite blocks. The highest type, mission blocks, are designed for an application and consist of a mix of other types of blocks. These blocks are based off of concepts from IEC61499 - Function Blocks for Embedded and Distributed Control Systems Design[2], with the addition of elementary blocks to reflect different design tools.

The logic blocks are tied to physical assemblies and products. Each mission block is linked to a specific product. Some, but not all, elementary, basic, and composite blocks get linked to a specific assembly or specific PCA. Other elementary blocks are linked to a specific design tool. By having a functional BOM structure, engineers can quickly navigate across designs for issues and reuse blocks for products and test fixtures. The test fixtures can be composed of blocks used in the systems that the assembly being tested mates in the actual products and common functional blocks specific to test fixtures. By continuing the physical BOM structure, products can be manufactured cheaply and with little confusion.

Common functional blocks should contain variants to accommodate specific applications while maintaining traceability.[3] The part numbering scheme or associated metadata should fully describe the block and accommodate addition or subtraction of features, so that descriptions are helpful but unnecessary. This will aid engineers in searching for blocks with specific attributes and will communicate precisely how these blocks are intended to interact with others.

A common functional block system can be illustrated with an example, such as the example RS-232 to TTL block described above, which is a good example of a basic block. The task of this basic block is to provide a translation of RS-232 signals from the outside world into the processor. The basic block consists of several elementary blocks: an elementary block of the schematic with an electrical specification, an elementary block of the software with a software block specification, and an elementary block of the layout.

Benefits of Common Functional Blocks

Common functional blocks can be reused and tracked across different designs, in both product designs and test fixtures. If the same block is already on another design in production, the block can be considered verified and design engineering time can be significantly reduced. The biggest benefit for design engineers is the case where a new design is required that can be completely designed with blocks already in production. Is this case, design engineering time is reduced significantly.

Variants of common functional blocks can be created to adapt them to different processors or different physical interfaces. In the RS-232 to TTL block example, the block is used in two locations, one location requires a DB-9 connector and the other requires a low profile connector. A variant is created and traceability is maintained.

Common functional blocks can be verified, tested, and revised just like any other design. In complex designs, engineers tend to speed through simpler circuits and focus on the more complex pieces of a design. The result is that mistakes in simple circuits can occasionally occur and are not caught until late in the design process. A benefit of breaking up a large design into smaller blocks is that it encourages engineers to consider every possible condition to which every block will be exposed. On new designs, engineers can borrow from previous designs with full knowledge that the design has already been fielded. If a problem is found on the block anywhere it's used, it can be traced back everywhere it's used.

Because block linkage is maintained, sustaining engineering becomes much easier. Instead of troubleshooting entire designs, problems are isolated to blocks. If a problem is found in the RS-232 example, engineers can instantly identify where else that block is used and implement solutions across all instances. This is a big benefit over the traditional copy and paste design engineering methods, where specific design knowledge about linkages is tribal knowledge that can be easily lost. Also, with the copy and paste method, any improvement made to the block have to be manually propagated back to all of the other designs from which they were copied.

Another benefit of common functional blocks is that they result in a lower total number of unique BOM items. Since the common functional block contains the same parts across several designs, it only has one BOM. In contrast, an RS-232 circuit that has been designed several times will contain different unique item numbers in each instance.

Common functional blocks can be used on both designs and tests. Functional test fixtures can be made with the same blocks that the units being tested may get exposed to in the field. Test fixtures can also benefit automatically from any sustaining activities performed on the products, including issues that can threaten production, such as obsolescence. Manufacturing engineers can create variants of common functional blocks with specific test objectives and fixtures can be developed to test multiple PCAs that contain similar sets of common functional blocks. The benefit of maintaining a common set of hardware across both product and functional fixtures is significant. Presently, a lot of time is spent supporting multiple functional test platforms. If manufacturing engineers currently support four functional test platforms and after transitioning to common functional blocks they support a single platform based off of products, significant efficiency gains will be achieved.

The biggest challenge for adapting common functional blocks to test fixtures will be the user interface. The use conditions associated with performing a test will be significantly different than the use conditions associated with controlling an advanced integrated system. There are a number of methods to solve this problem.

One method is to find an off-the-shelf industrial control platform or test development platform. Industrial control platforms are highly modular and easily adaptable to a variety of physical interfaces, specifically analog, digital signals, serial signals, and Ethernet. Automated JTAG (Joint Test Action Group) test development systems can import net lists or ODB++ data on both products and test fixtures and automatically develop comprehensive tests without a single line of code ever being written.

Another method is to extend common functional blocks to engineering development software. At our company, an entire library exists to interface a computer to specialized hardware. The library consists of a variety of C# classes and structures used to develop test applications. This library can be adapted to interface with both production hardware and test fixtures based off of production hardware. A common user interface can be created and then routines can be written in C#.

If the controls software runs in commercial PCs, it will inevitably be created with a variety of standard interface modules that could be ported to custom test applications. A standard test user interface could become a common functional block, and then specific code would be developed to tie the test block to the product code.

An automated JTAG development system, combined with custom test hardware that contains a JTAG device and augmented with occasional test applications, can provide thorough testing with minimal development time and platform maintenance. The automated JTAG system can place both the unit under test and the test fixture hardware into JTAG mode and get comprehensive circuit test coverage through connectors. Then, the system can put the unit under test into functional mode where it runs with product software. The test fixture would still be in JTAG mode and be controlled by the automated JTAG system, so no custom test code is required for the test fixture JTAG device.

For specialized tests, an interface mounted in the test fixture (such as a commercial CAN to USB device) can provide a data stream from the unit under test to a PC. Then a software interface developed for product can interface with a common test user interface block. If the automated JTAG system has a software developer's interface, it can also interface with a common test user interface block. The end result is complete control and testing of the assembly using hardware and software found in product with a simple, common user interface and minimal time developing code unique to a particular test fixture.

Examples of Benefits

Functional blocks have benefits for design engineering, sustaining engineering, and manufacturing engineering groups. For examples of benefits, the RS-232 to TTL block described earlier will be looked at from the system perspective and from the product lifecycle perspective.

This RS-232 to TTL basic block is part of the vehicle motor control composite block, which has the tasks of communicating with the control station and coordinating motor controls. This composite block consists of several other composite, basic, and elementary blocks. It has several motor control basic blocks, each of which has the task of sending speed and position commands and power to a single motor. It also has the same amount of motor feedback basic blocks, each of which has the tasks of receiving, filtering, and interpreting speed and position feedback. It has one power supply basic block, which has the task of providing clean power to the motors, sensors, and controls, and one processor basic block, which has the task of communicating with the control station and coordinating motor control. Finally, it has various other blocks to perform any other actions required to accomplish the task of controlling motors.

The vehicle motor control composite block is part of the system motors composite block, which also includes a control station vehicle movement composite block; motor basic blocks; and a mix of other elementary, basic, and composite blocks. The vehicle motor control composite block is part of the vehicle mission block.

The vehicle motor control composite block is a single physical assembly and is linked to the physical assembly part number. Any sustaining, troubleshooting, or testing activities related to this physical assembly are assisted by the functional assembly. For example, when a sustaining engineer troubleshoots the assembly and suspects that the design of the power supply basic block is causing intermittent brownouts under specific conditions, he can instantly see everywhere else that power supply block is used. This additional information enables him to see if the power supply basic block exhibits the same behavior in the other locations or if something else is causing the brownout.

In this example, the engineer determines that the brownouts are caused by a defect in the power supply basic block and triggered by specific behavior unique to systems with high inrush currents. This block is used in two other places: a sonar composite block and a camera composite block. The sonar composite block has high inrush currents and the engineer finds that under very rare circumstances, the problem can affect the sonar composite block as well. The camera composite block does not have high inrush currents. However, the change is an improvement on the robustness of the design and has little impact on cost. The engineer releases the new revision of the power supply basic block and triggers immediate updates to the vehicle motor control composite block and the sonar composite block. The revision is queued for the camera composite block, so that one year later, when the camera composite block is revised due to planned obsolescence activities, the block also gets the power supply basic block improvements.

In another example, the company decides to add BIST (built in self-test) across all products for more thorough, low cost production testing.[4] There are 75 PCAs and 16 basic schematic blocks, each with several variants. Because all of the PCAs are part of a functional common functional block system, engineers can quickly break up the task into small pieces. The processor basic block is modified to contain a processor with a higher pin count to accommodate new input and output (I/O) requirements of BIST and a common interface is defined. The work of modifying the other schematic blocks is divided among the other engineers. When the first assembly with BIST is built, the engineers find an improvement to the RS-232 to TTL basic block implementation of BIST and modify the block again. Test fixtures across all PCAs need to be modified to test the new functionality. By the time engineers have touched a third of the PCAs, each of the 16 main basic schematic blocks has been updated and verified. Updating the remainder of the PCAs goes very quickly, as engineers spend little time on design of products or new fixtures. The remainder of the work involves checking that the changes will work with the various combinations and variants and then re-qualifying the assemblies.

Commercial Functional Test Platform Deficiencies

Common functional blocks benefit in house designed functional test fixtures. This section offers an explanation why many existing commercial functional test platforms are inadequate in a low volume, long life systems production environment. Some of the deficiencies with many commercial fixtures are related to fixture costs per product, specialized overhead required to support test platforms, performance variations between test equipment and products, and fixture obsolescence.

Fixture costs come from two primary sources: materials costs and time. Many commercial test fixtures run on a base that can cost tens of thousands of thousands of dollars or more and may still require significant customization to make the fixtures operate appropriately with a specific product. While the speeds at which these fixtures operate may justify the costs in high volume applications, fixture costs can exceed total hardware costs by several times in low volume applications.

A significant portion of production test cost comes from engineering time required to develop and maintain fixtures. Materials cost can be minimized by using open source hardware for test fixtures, but engineering time spent to develop and maintain these fixtures can be significant. The lack of fixture controls common in lower volume production environments frequently means that the engineer who originally designed the fixture is the only person who can troubleshoot it expediently. This is due to differences in documentation, selections of different platforms, and variations in test design methods. Because test fixtures are either completely custom or developed using reuse, test modifications to support new product revisions take a lot of time. By adopting common functional blocks, test fixtures will benefit from the same controls as designs. Manufacturing engineers already have to learn product design specifics to create test fixtures. By using the same blocks as the designs, the requirement to retain knowledge of platforms exclusive to testing is reduced and resources can be more appropriately balanced among testing needs.

By developing test fixtures with platforms different than what products encounter in the field, performance variations are introduced. Test fixtures also generally don't get the same level of design scrutiny that products enjoy. These variations combined with test fixture wear can lead to fixtures failing or passing boards erroneously.

Most importantly, any platform (both for products and fixtures) has a risk of obsolescence. Serious sustaining engineering efforts are employed when an obsolescence issue hits a product. Obsolescence issues with test fixtures can have serious impacts when the fixture breaks down or needs modifying. If the fixture can not be made to work, production stops. Millions of dollars are spent and entire business models are based on test fixture obsolescence, including companies that go through the extraordinary step of reverse engineering fixtures to meet program requirements.[5] This problem is avoided by having a test fixture development program that uses product blocks as the basis for test fixtures. For integrated systems manufacturers, most of the components needed for test fixtures are already designed in their products.

Process Changes and Technical Hurdles

Adopting a common functional block scheme is a major shift in product development that will require changing processes and overcoming technical hurdles and organizational challenges. It is important that affected employees understand the motivation and that management strongly endorses the concept prior to initiating the changes.

Process Changes
Common functional blocks require process changes in several areas: documentation, sustaining engineering, design engineering, and manufacturing engineering. Common functional blocks also require a hierarchical documentation scheme, similar to physical BOMs. In most organizations, product documents are tied to specific physical assemblies, but are not linked to each other. A documentation scheme should be developed that allows maximum functional block flexibility and maximum flexibility in linking functional blocks to physical parts. The documentation scheme should allow for different revisions of the same blocks to exist at the same time in order to minimize and control product revisions. At the same time, an interface must be developed that allows users to move back and forth between functional and physical design hierarchies with ease.

Employees will need to be trained on the new documentation scheme and encouraged to use it. Sustaining engineers will need to understand how tracking linkages will help them track problems common to multiple assemblies. They will also need processes in place to capture issues found in blocks and plan revisions accordingly. Design engineers will need to be trained on how to quickly search and use blocks on new designs and will need to be aware of the benefits of creating new blocks.

In addition to familiarizing themselves with the new processes and block documentation, manufacturing engineers will need to establish standards for using and interfacing blocks in test fixtures. This will likely include the development of a common user interface. Manufacturing engineers can either develop a common protocol to communicate data from the production test fixtures with product designs to a computer program, or they can use product computer software and change the user interface portion. At our company, engineers have already developed a software interface library for development test applications. This library could also be used for developing production test applications.

Technical Hurdles
There are a few technical hurdles involved when implementing common functional blocks that will need to be resolved. The primary hurdles are related to highly integrated designs, ERP (enterprise resource planning) system capabilities, and interfacing blocks with engineering tools.

Highly integrated designs do not always appear to lend themselves to common functional blocks from the outset, particularly at the layout level. Common functional blocks can still be implemented, just with larger basic blocks. For example, designs centered on a CPLD will have a power supply block, memory blocks, and other I/O blocks. In extremely dense designs, it is not always practical to use default layouts. In these cases, variants may be used or the common functional block can exclude the layout, which can consist of a separate elementary block.

ERP systems may not be easily extendable to common functional blocks. These problems will be specific to the ERP system in place and will have to be resolved with the information technology, document control, and engineering departments.

Engineering tools do not always lend themselves to simple integration into a wider functional part control system; for example, many tools have their own vault systems. Careful planning can result in pieces that can easily be imported and exported from tools, such as schematic pages and software header files. The levels at which elementary blocks are created may be affected by tool limitations, but the elementary block's primary purpose is to accommodate these limitations.

The scope of the technical hurdles will vary widely by organization. With proper planning and support, each of these hurdles can be overcome.

A Phased Implementation Approach
Shifting to a common functional blocks architecture can be one large endeavor or a series of small projects. Given the need for normal business to continue during any transition, it is recommended to pursue a phased implementation approach. New designs at manufacturing companies are typically developed under severe time constraints in order to hit a specific market window. For this reason, it would benefit organizations to initially roll out common functional blocks during a major sustaining effort on an existing product. After it becomes apparent that existing designs are easy to document with functional blocks and that the major process issues have been resolved, pushing common functional blocks out to new product development would likely be met with minimal resistance. In addition, new product development would be equipped with a library of blocks to start building from, resulting in shortened design cycles from the outset of common functional blocks usage.

Selecting the first product to transition into using common functional blocks should be done carefully. It should be a product that has major upcoming sustaining efforts, so that any common functional blocks effort can be integrated into normal business activities with minimal disruption to resource planning. Once the product is selected, major subsystems can be blocked out and a skeleton functional hierarchy should be documented.[6] It's important that the hierarchy follow function and not physical location so that the blocks can be reused. With a draft top level hierarchy in place, work on a single PCA or single assembly can commence.

Implementing common functional blocks should also encourage engineers to find efficiencies in circuits. For example, engineers may find that the same serial interface is implemented differently in different parts of the design, both in hardware and in code. By creating a serial interface block with a common specification, physical BOMs will consequently change and unique BOM item count will decrease.

Plenty of extra time should be allotted for conversion of the first assembly into common functional blocks, as additional design and documentation efforts, coupled with uncovering inevitable process glitches, will take more time than initially. A company should pursue the usage of common functional blocks with the full knowledge that the first few will take longer than average and that the time advantages will be achieved after the first few PCAs are developed with the new architecture and remaining process issues are resolved. Any significant time advantages will be realized over the life of the products. Because test fixtures are designed with the same blocks, any specialized test hardware could be developed on a much quicker cycle, assuming the manufacturing engineer is familiar with the underlying product technologies.

Conclusion

Implementing common functional blocks will benefit new product development, sustaining engineering, and manufacturing engineering groups. Design engineering groups will benefit from faster time to market on new designs and decreased unique BOM item count. Sustaining engineering groups will benefit from traceable, robust designs and the ability to improve several designs at once by examining blocks. Manufacturing engineering groups will benefit from having fewer test platforms to support and being able to use tools and resources that are traditionally exclusive to product development. Common functional blocks will significantly reduce total product lifecycle costs.

References

[1] Vidojkovic, M.; van der Tang, J.; Baltus, P.; van Roermund, A., "Design of Flexible RF Building Blocks-A Method for Implementing Configurable RF Transceiver Architectures," *Communications, 2005 Asia-Pacific Conference on*, Oct. 2005

[2] Feng Xia; Zhi Wang; Youxian Sun, "Function block based design pattern for flexible manufacturing control system," *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, Oct. 2003

[3] Bair, O.; Soundararajan, S.; Chen, A.; Carbonara, M.; Joly, C., "System design environment with synthesized blocks approach," *ASIC Conference and Exhibit, 1995., Proceedings of the Eighth Annual IEEE International*, Sep 1995

[4] YaJun Gu; Ye Qin; ZhiJun Wang; David Wei; Andrew Ho; Stephen Chen; Zhen (Jane) Feng Ph. D.; Murad Kurwa, "Application Of Build-in Self Test In Functional Test Of DSL," *IPC APEX EXPO*, 2012

[5] Sarfi, T., "COTS VXI hardware combats obsolescence and reduces cost," *IEEE AUTOTESTCON Proceedings*, 2002

[6] Wen-Tin Lee; Kuo-Hsun Hsu; Lee, J., "Designing Software Architecture with Use Case Blocks Using the Design Structure Matrix," *Computer, Consumer and Control (IS3C), 2012 International Symposium on*, June 2012

Reduce Design Time and Product Lifecycle Costs with Functional Blocks Common to Designs and Test Fixtures

Stephen Golemme
FMC Technologies Schilling Robotics
Stephen.Golemme@fmcti.com

Summary

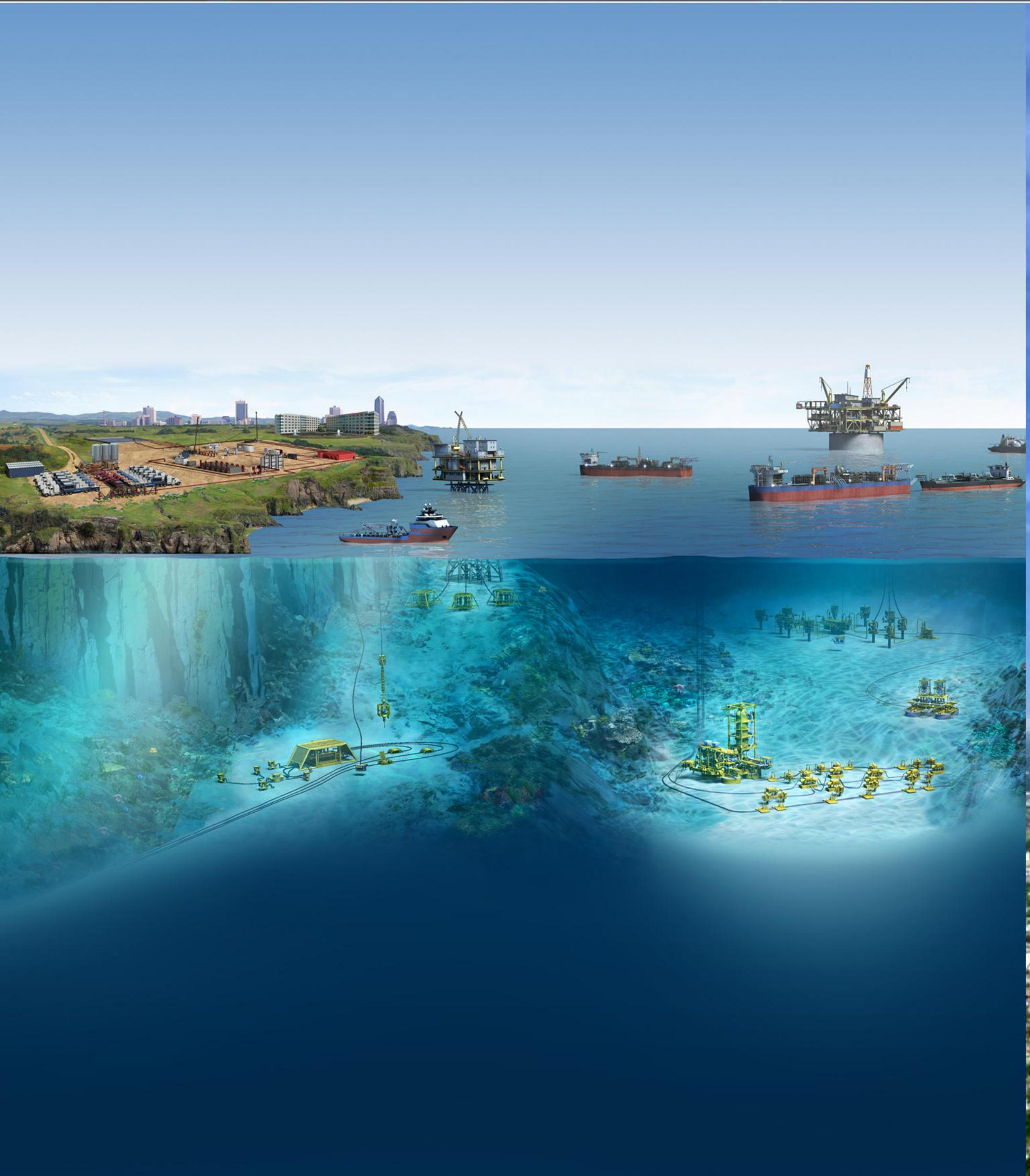
- Common Functional Blocks Applied:
  - Across designs
  - Across disciplines
  - To both products and fixtures
- A platform that provides for:
  - Variants
  - Versions
  - Traceability
- Benefits:
  - Save time
  - Improve quality
  - Find and fix errors quickly
  - Focus on function

Agenda

- Context, history
- Block levels
- Functional platform-based block structure
- Benefits:
  - Design benefits
  - Sustaining benefits
  - Functional test benefits
- Updates

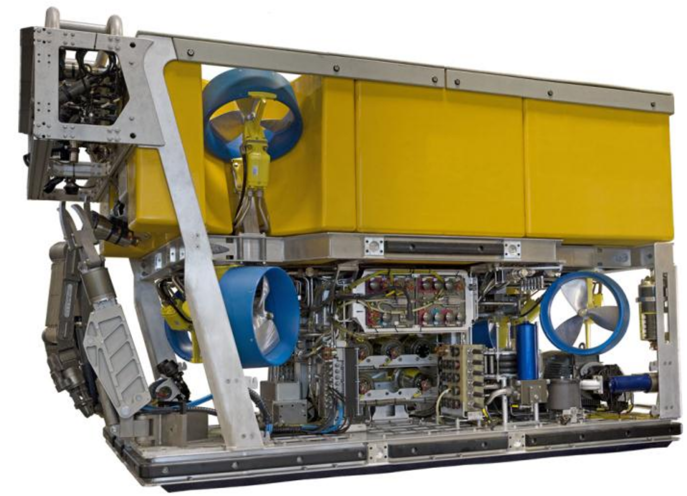
Agenda

- **Context, history**
- Block levels
- Functional platform-based block structure
- Benefits:
  - Design benefits
  - Sustaining benefits
  - Functional test benefits
- Updates

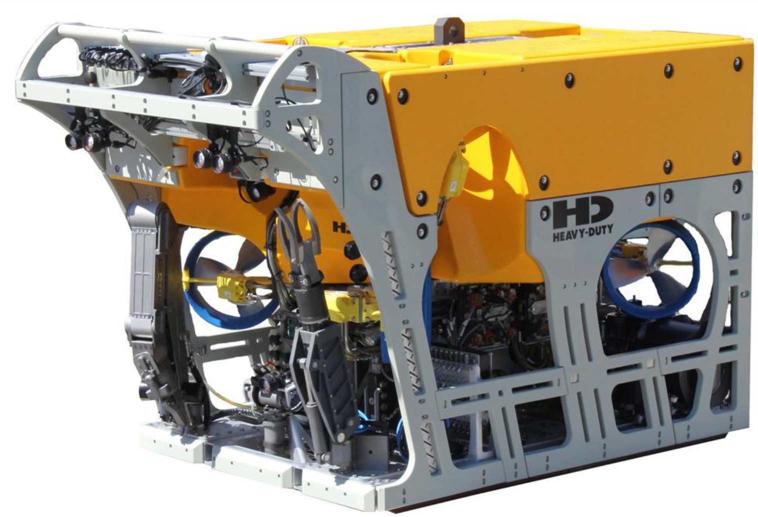


Context

3,000+ Manipulators Delivered



100+ Work-Class ROVs Delivered



Common Blocks – Modern History

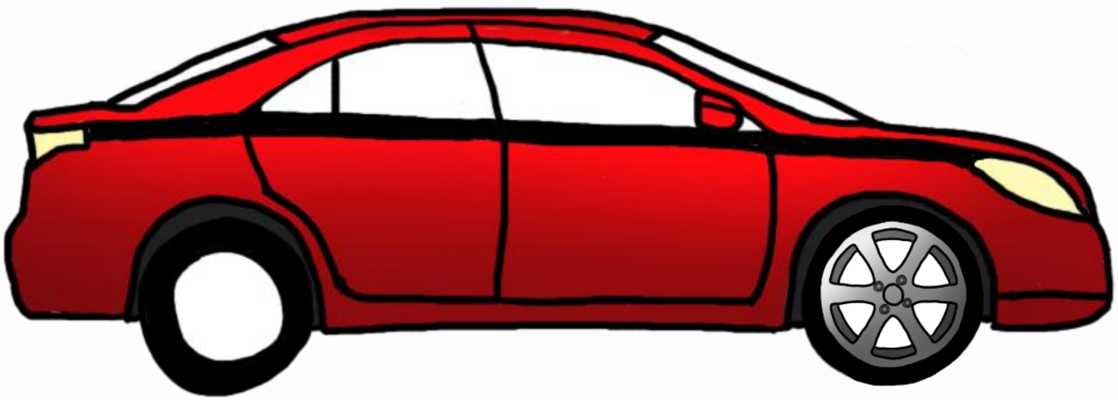
- Software
- Industrial control systems
- ASIC design
- RF design
- Others…

Agenda

- ✓ Context, history
- **Block levels**
- Functional platform-based block structure
- Benefits:
  - Design benefits
  - Sustaining benefits
  - Functional test benefits
- Updates

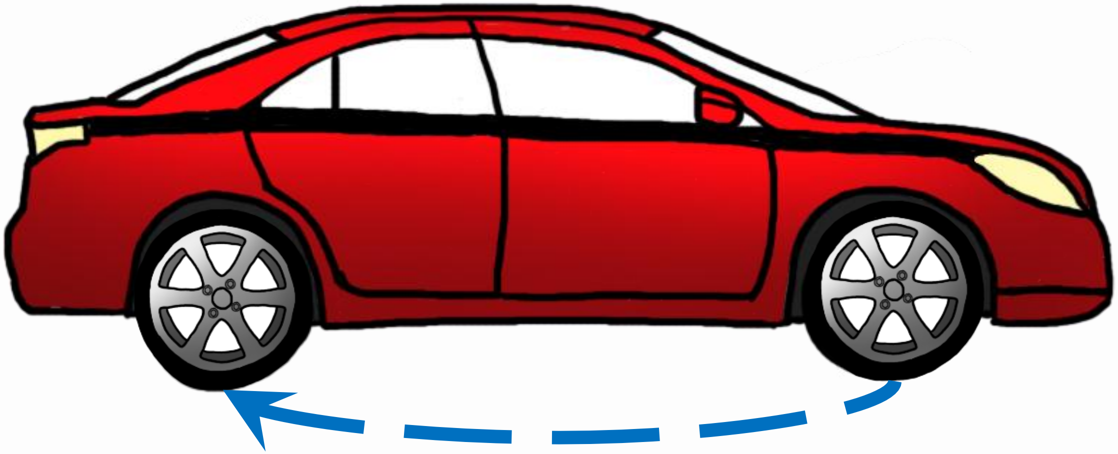
Common Block Levels

- Custom
  - Most time



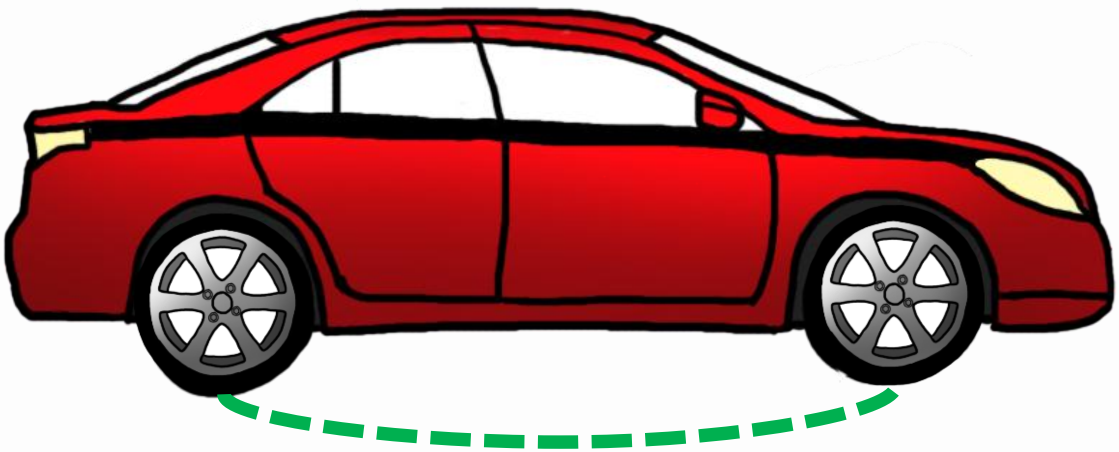
Common Block Levels

- Custom
  - Most time
- Reuse
  - Copy & paste



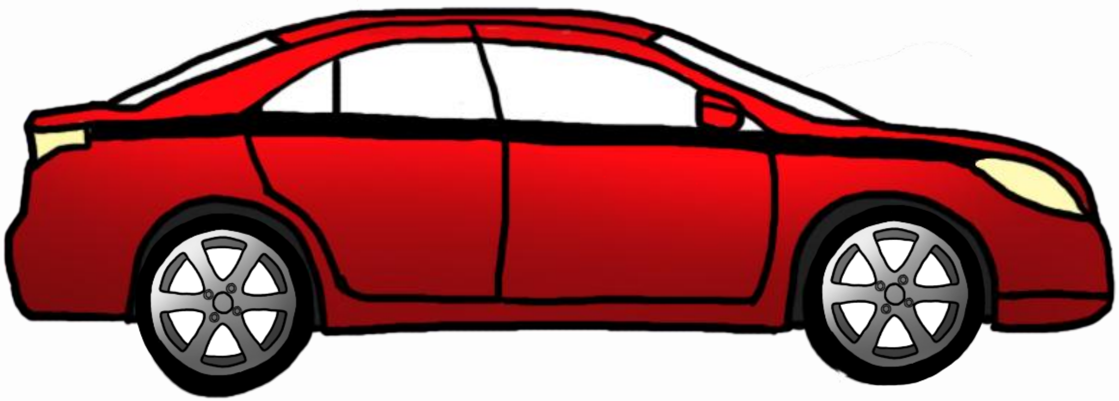
Common Block Levels

- Custom
  - Most time
- Reuse
  - Copy & paste
- Platform
  - Traceable blocks



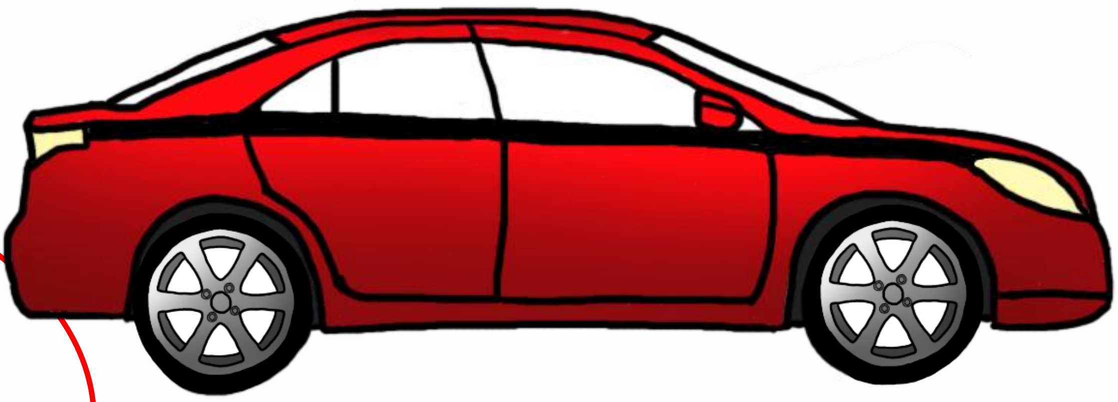
Common Block Levels

- Custom
  - Most time
- Reuse
  - Copy & paste
- Platform
  - Traceable blocks
- Reconfigurable
  - $$$



Common Block Levels

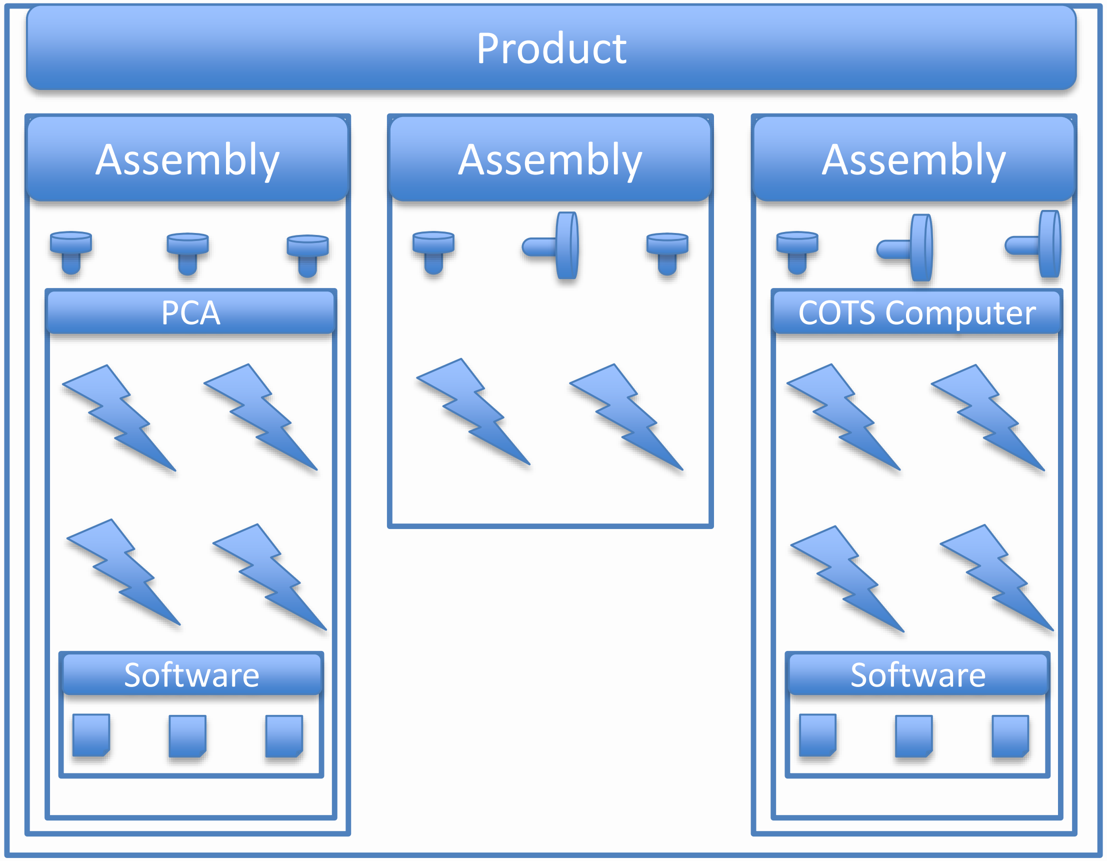
- Custom
  - Most time
- Reuse
  - Copy & paste
- Platform
  - Traceable blocks
- Reconfigurable
  - $$$



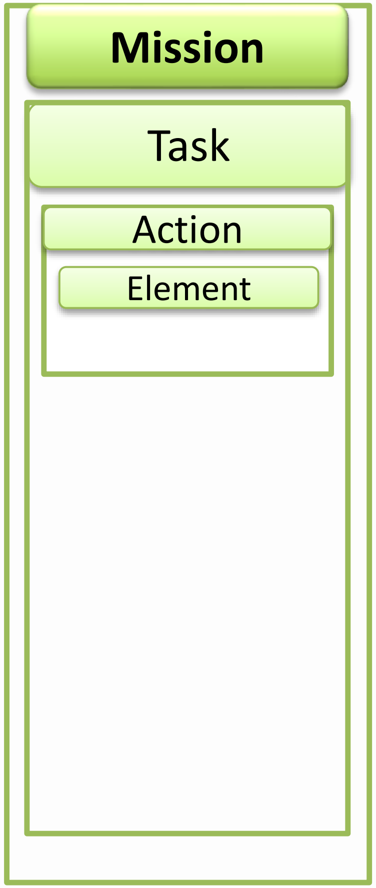
Agenda

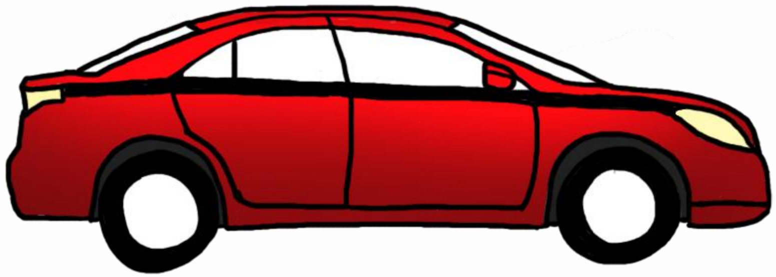
- ✓ Context, history
- ✓ Block levels
- **Functional platform-based block structure**
- Benefits:
  - Design benefits
  - Sustaining benefits
  - Functional test benefits
- Updates

Manufacturing BOM Structures



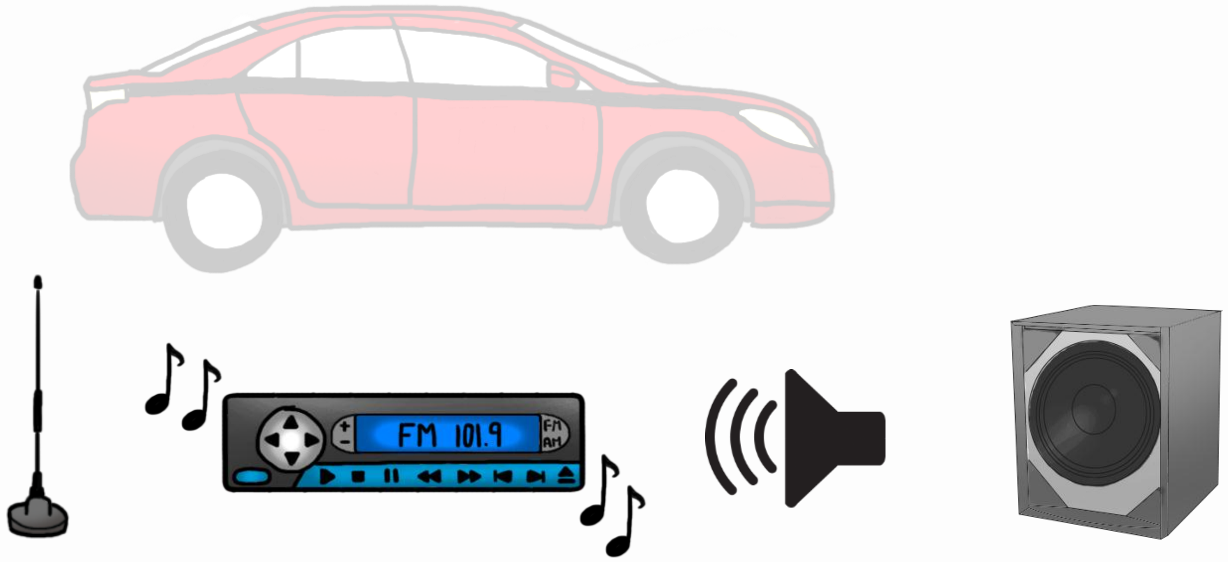
Proposed Functional BOM Structure



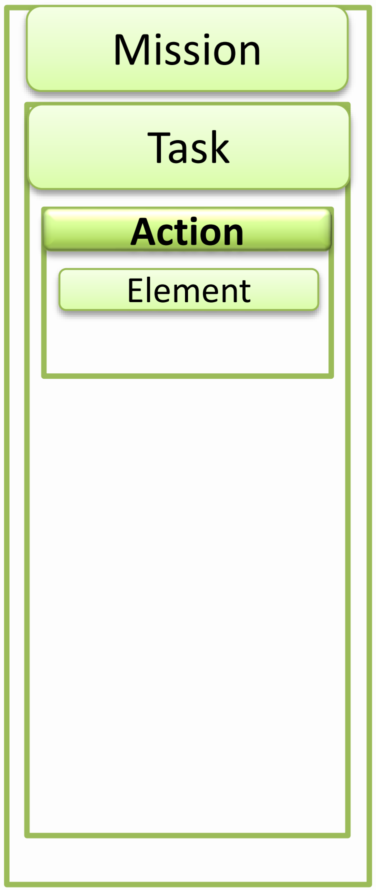


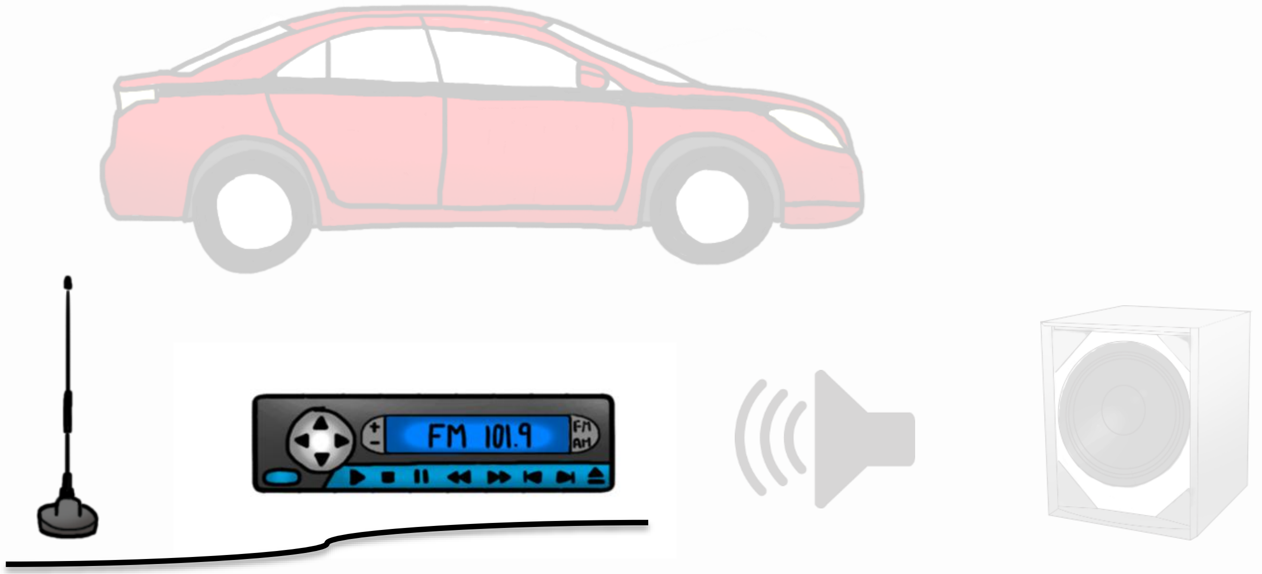
Proposed Functional BOM Structure

- Mission
  - **Task**
    - Action
      - Element

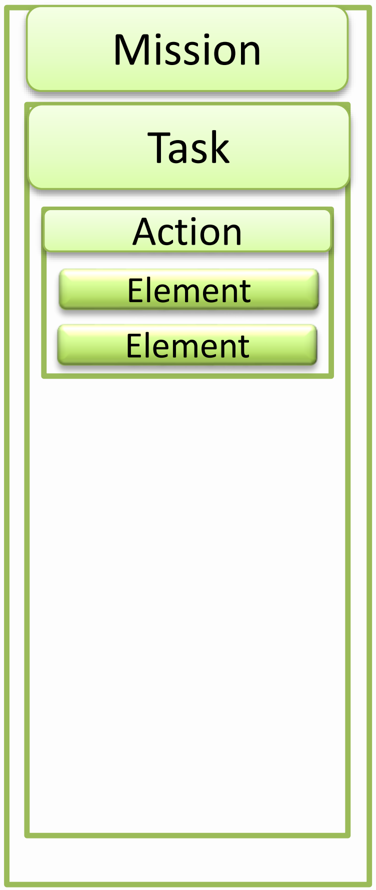


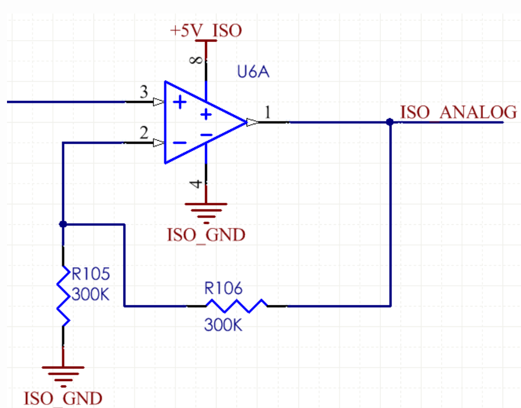
Proposed Functional BOM Structure

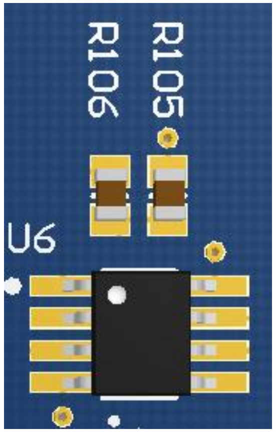




Proposed Functional BOM Structure

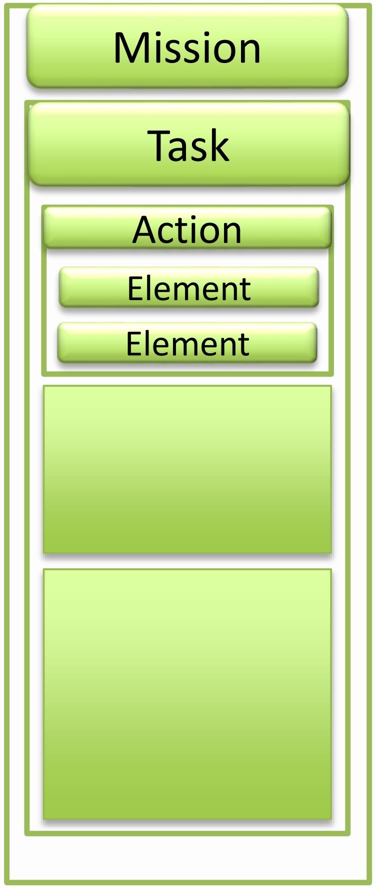


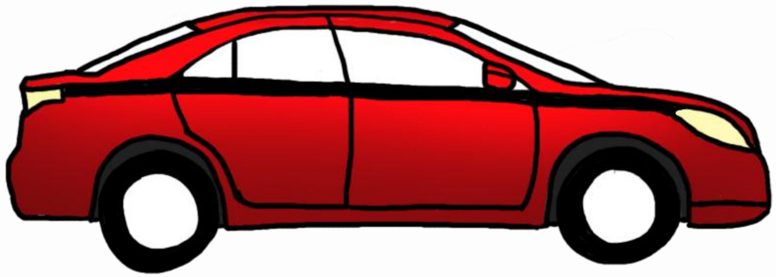


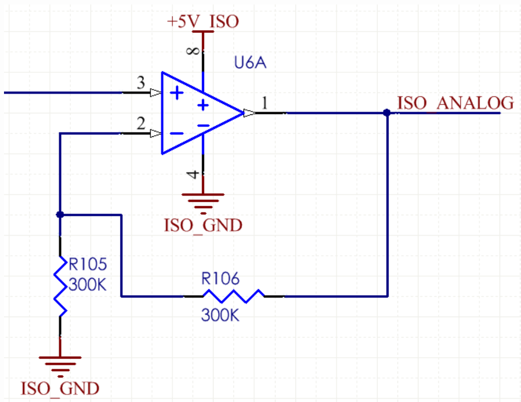


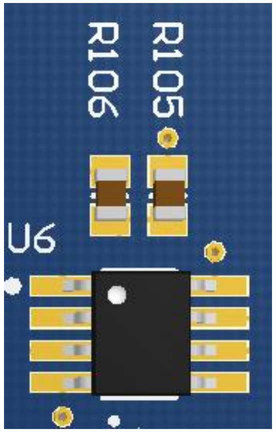

```
for (int i = 0; i < data.Length; i++)
  {
     v0 =  (1 - a * c) * v0  -  (c) * v1  + (c) * data[i];
     v1 =  (1 - a * c) * v1  +  (c) * v0;
     data[i] = v1;
  }
```

Proposed Functional BOM Structure





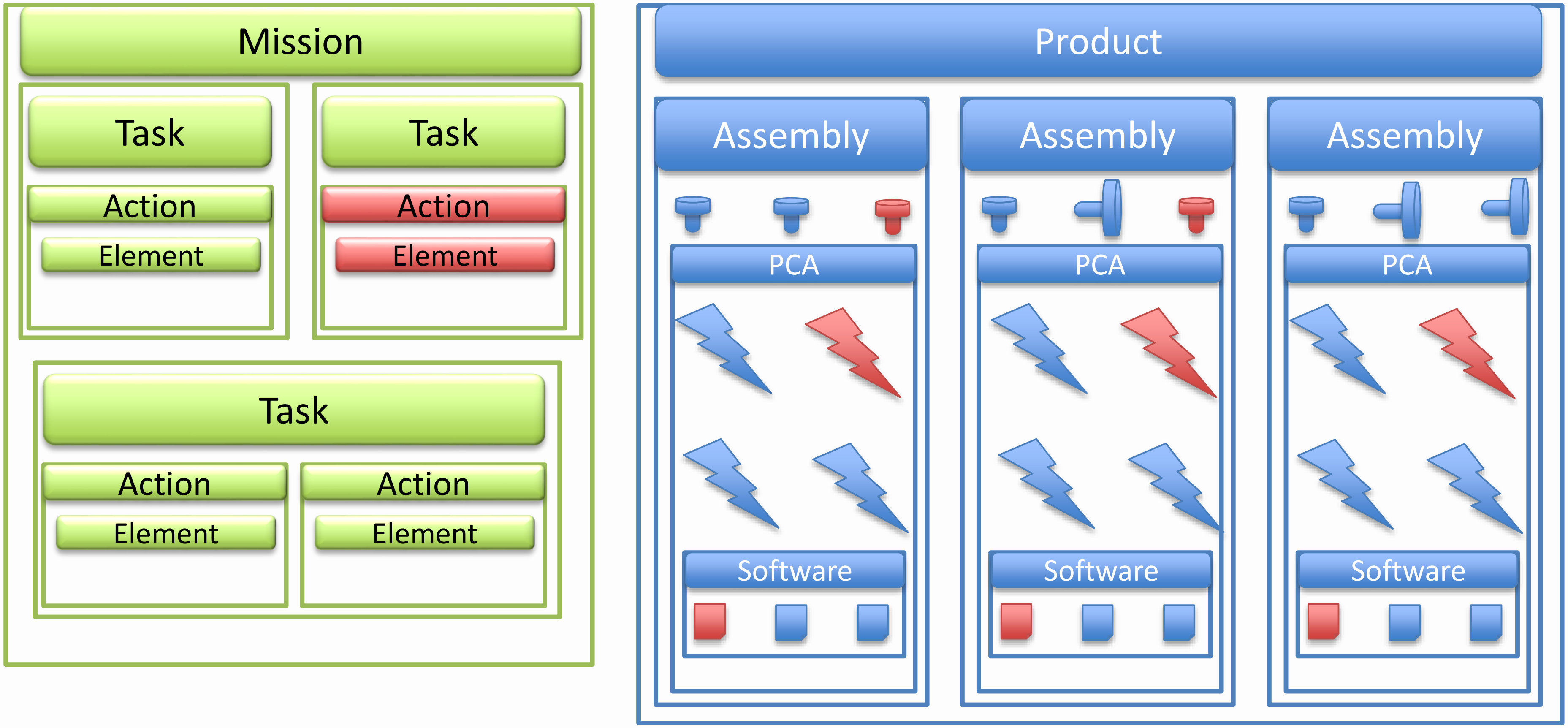




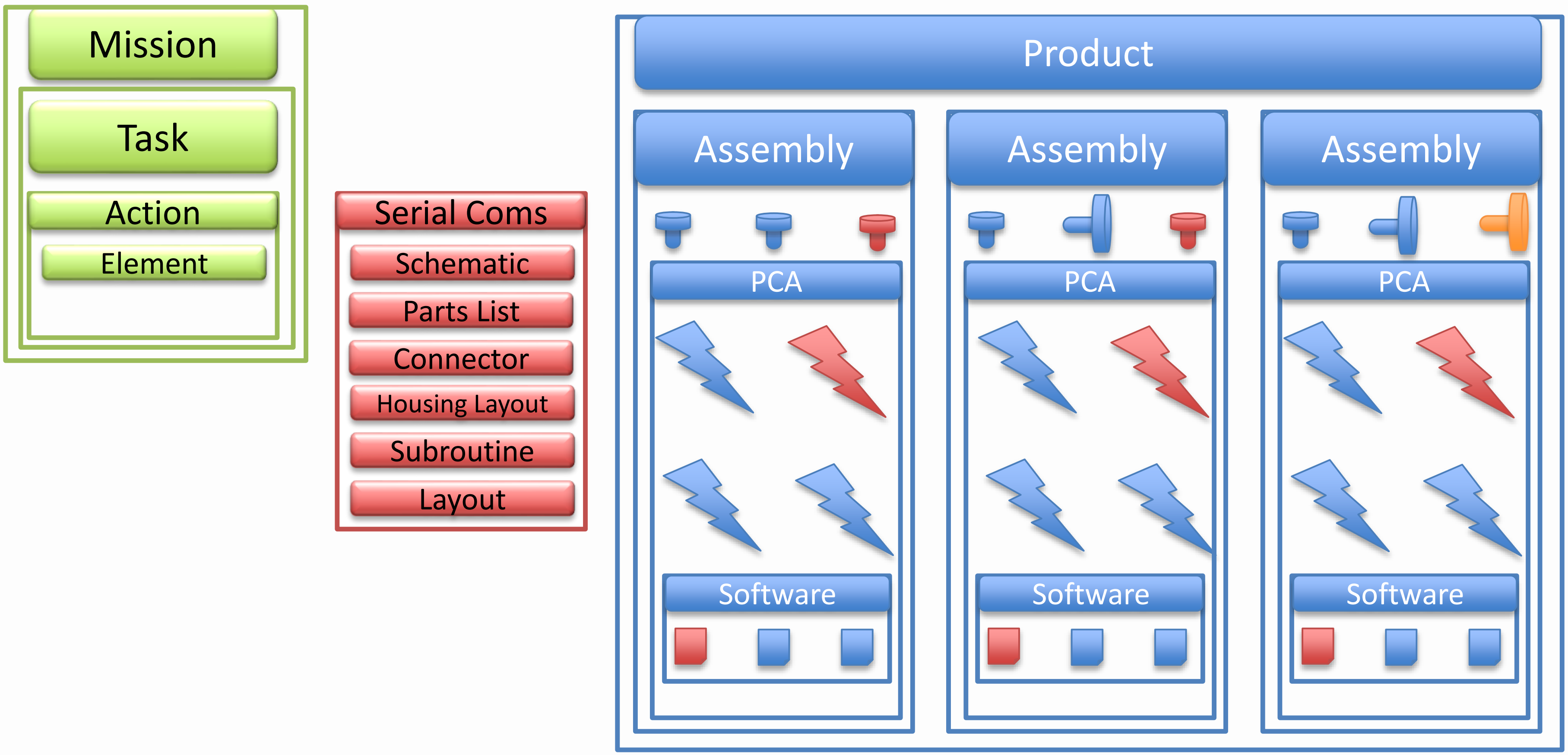

```
for (int i = 0; i < data.Length; i++)
  {
     v0 =  (1 - a * c) * v0  -  (c) * v1  + (c) * data[i];
     v1 =  (1 - a * c) * v1  +  (c) * v0;
     data[i] = v1;
  }
```

Linking BOM Structures

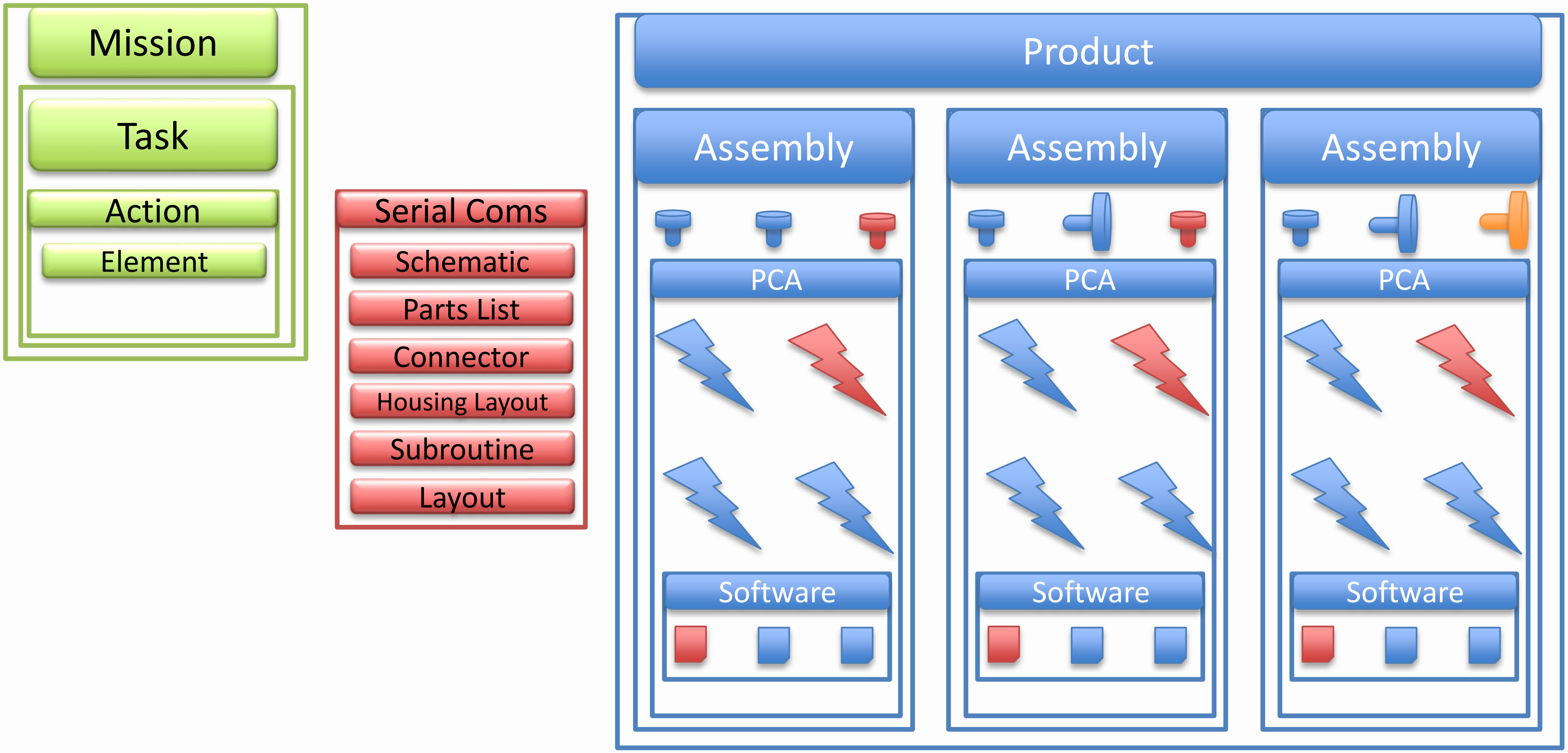
Functional **Physical**



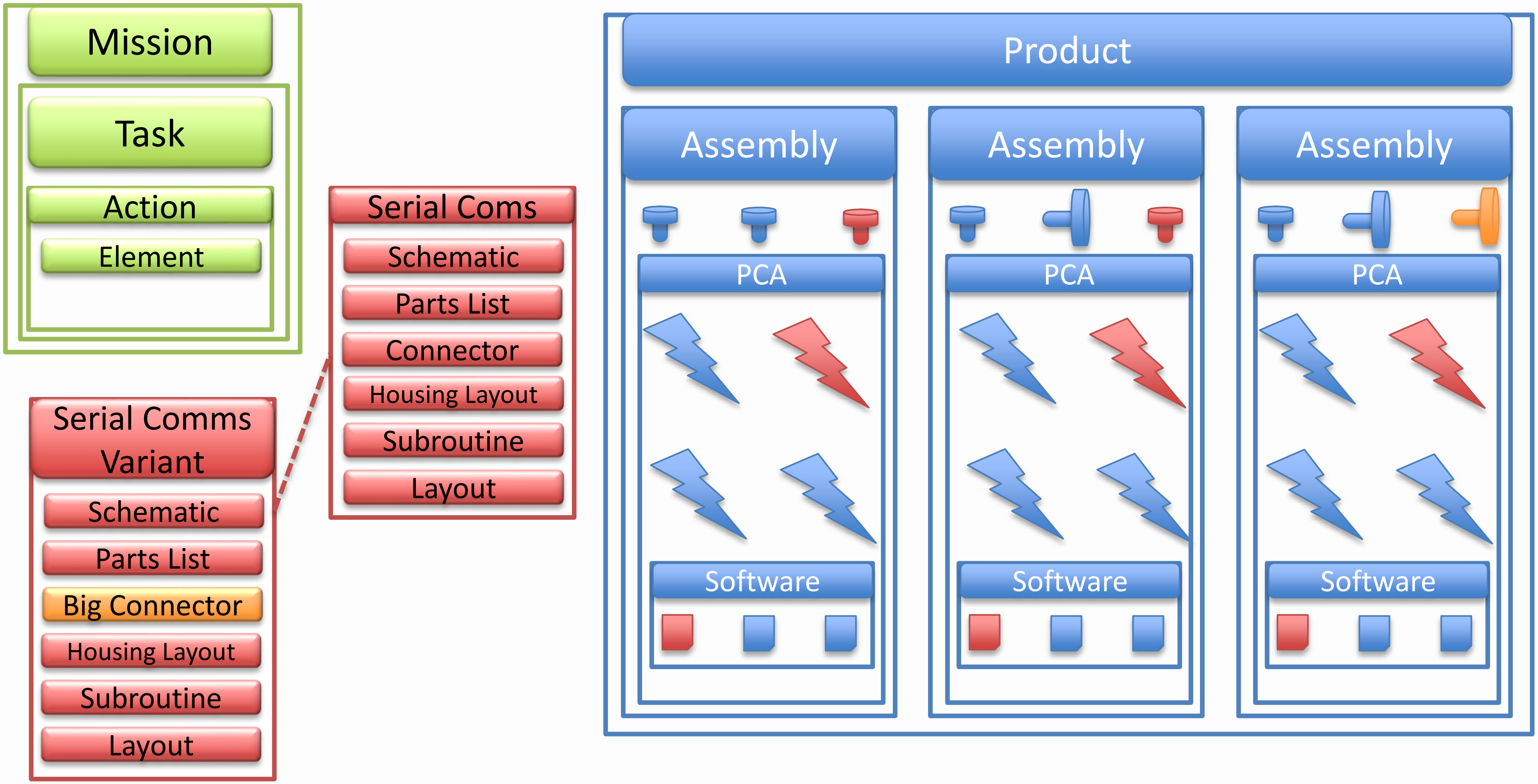
Action Block



Functional Variants



Functional Variants



Platform vs. Copy & Paste

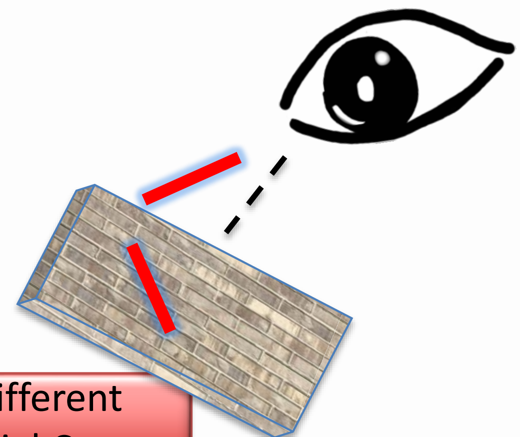
Platform

Serial Coms
- Schematic
- Parts List
- Connector
- Housing Layout
- Subroutine
- Layout

Serial Coms Variant
- Schematic
- Parts List
- Big Connector
- Housing Layout
- Subroutine
- Layout

Copy & Paste

Serial Coms
- Schematic
- Parts List
- Connector
- Housing Layout
- Subroutine
- Layout

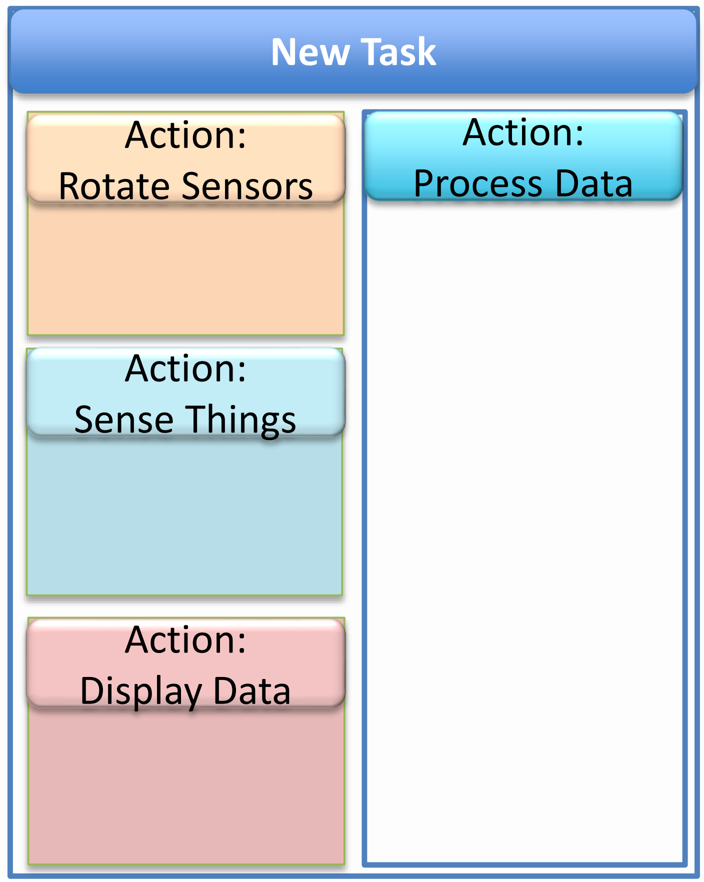


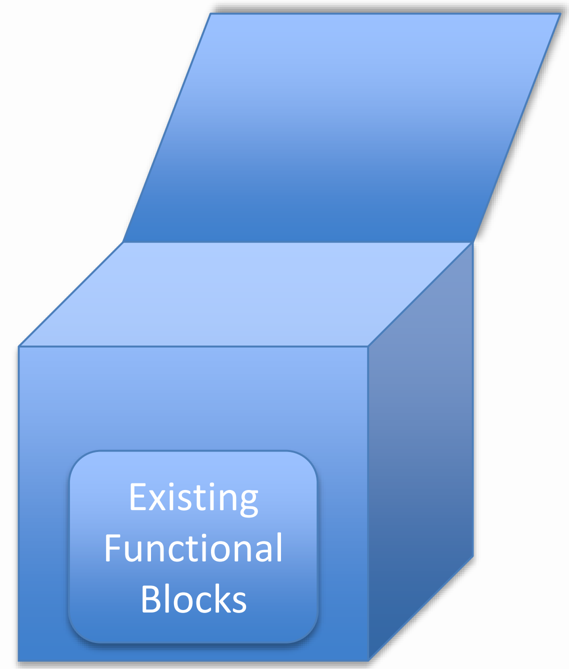
Different Serial Coms
- Schematic
- Parts List
- Big Connector
- Housing Layout
- Subroutine
- Layout

Agenda

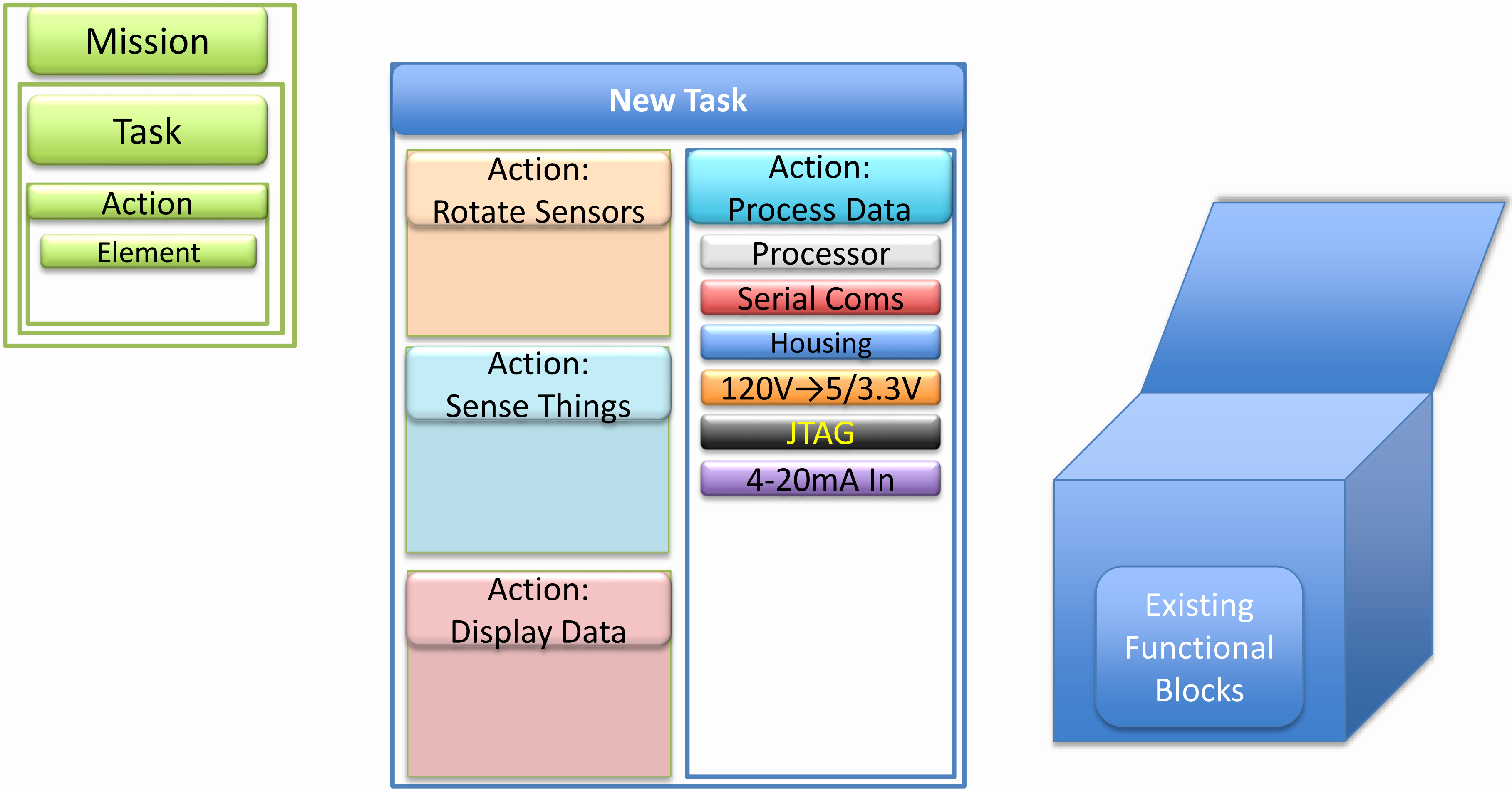
- ✓ Context, History
- ✓ Block Levels
- ✓ Functional Platform-Based Block Structure
- Benefits:
  - **Design Benefits**
  - Sustaining Benefits
  - Functional Test Benefits
- Updates

Faster Design

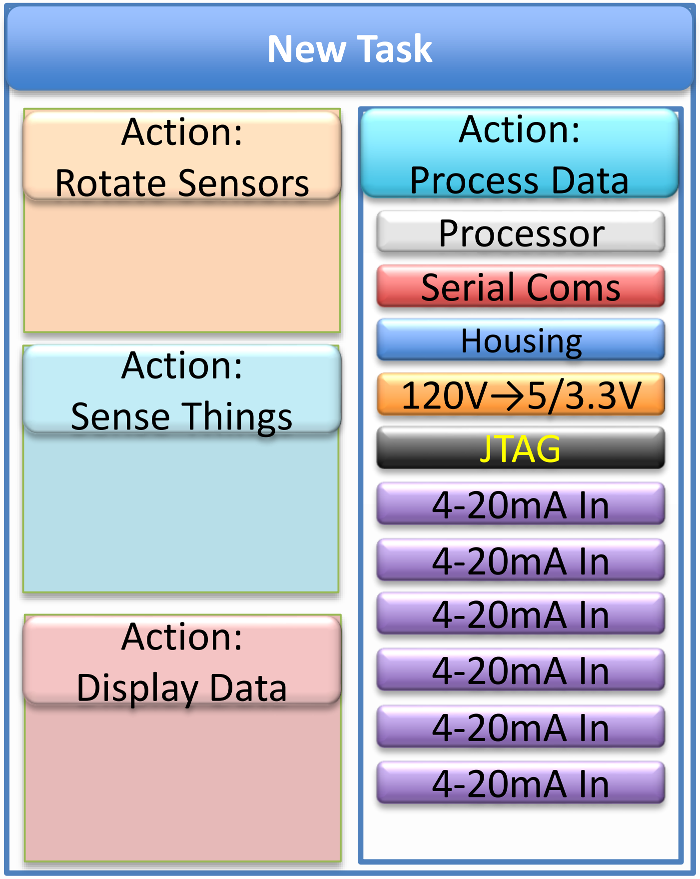


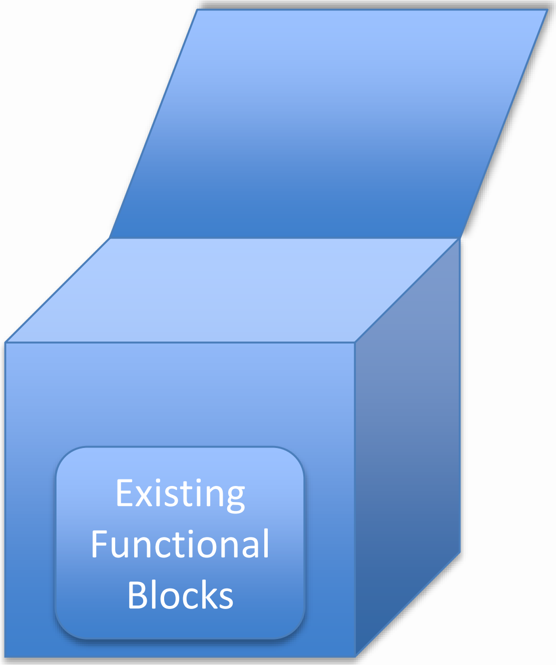


Faster Design

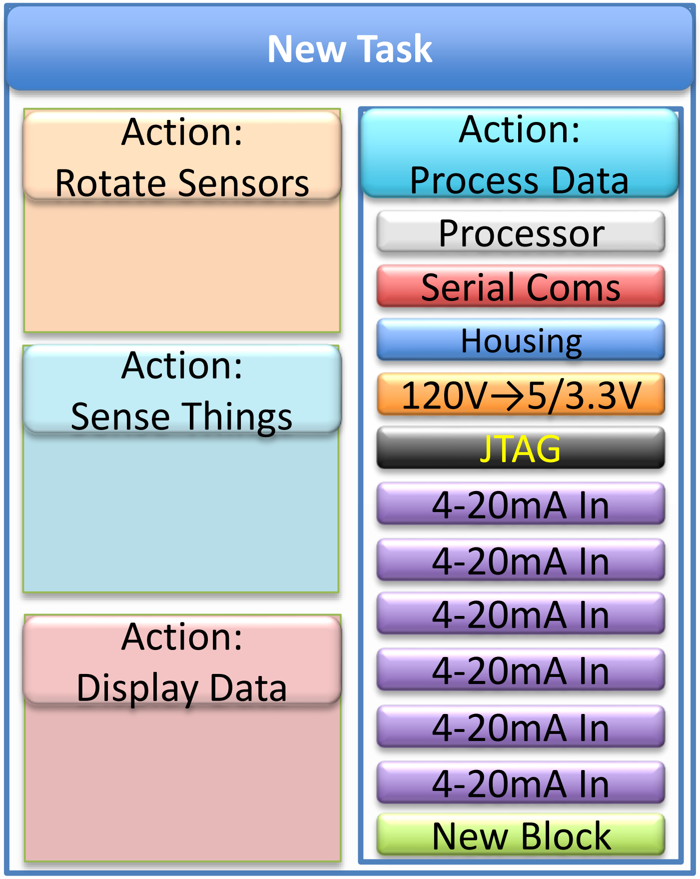


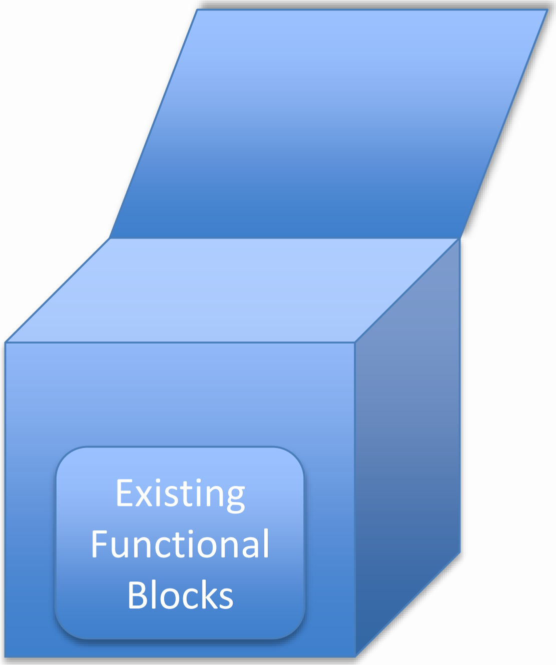
Faster Design





Faster Design

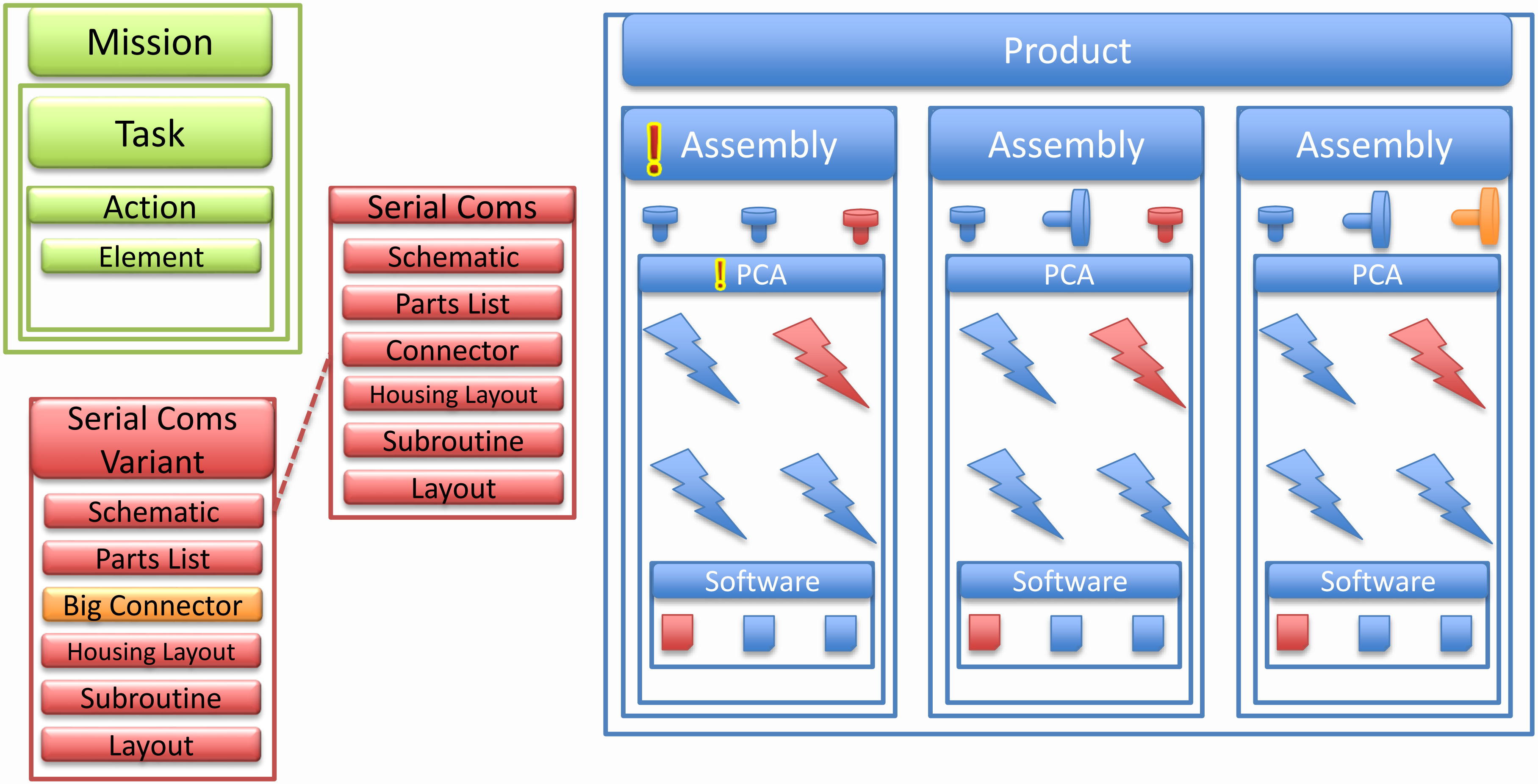




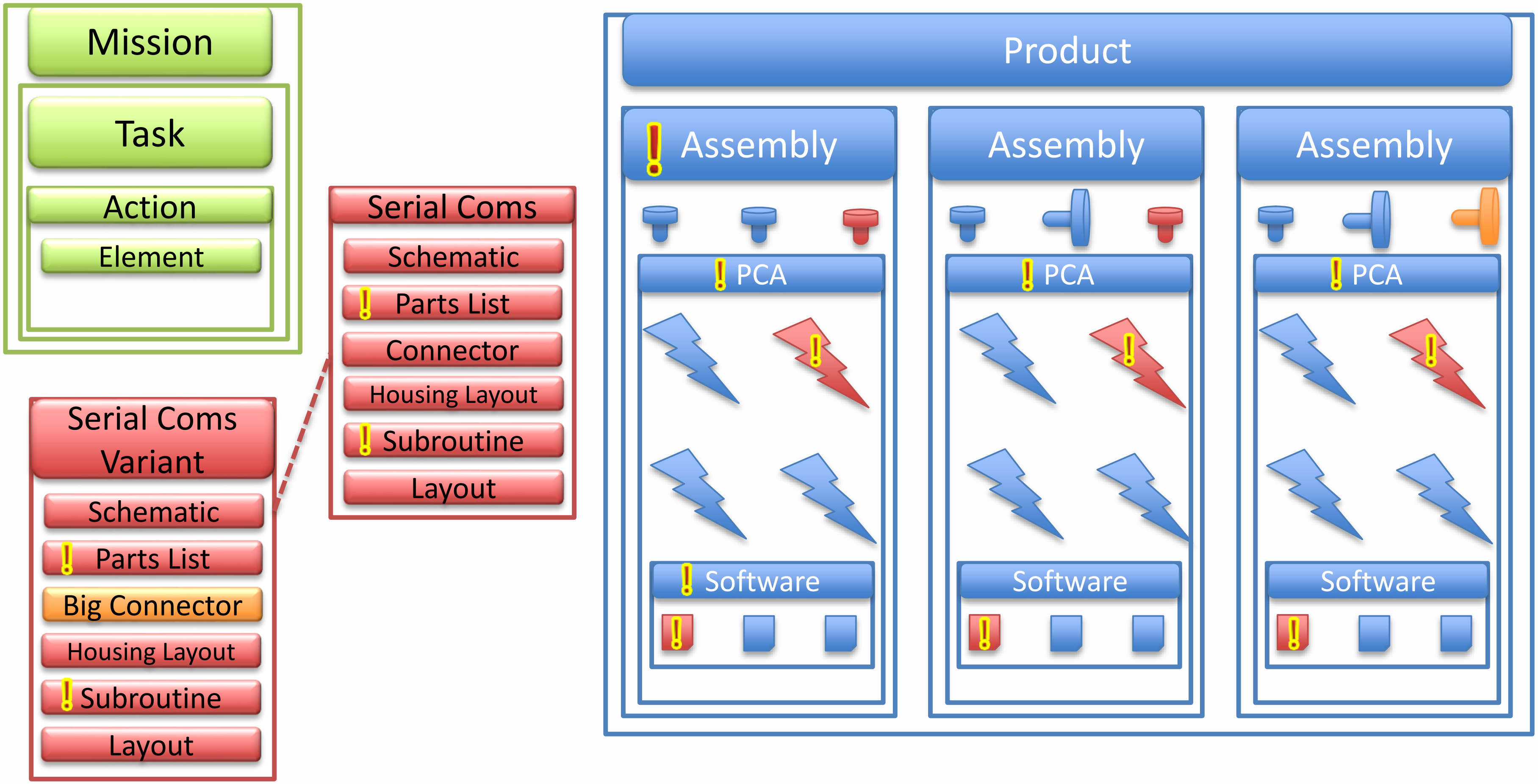
Agenda

- ✓ Context, History
- ✓ Block Levels
- ✓ Functional Platform-Based Block Structure
- Benefits:
  - ✓ Design Benefits
  - **Sustaining Benefits**
  - Functional Test Benefits
- Updates

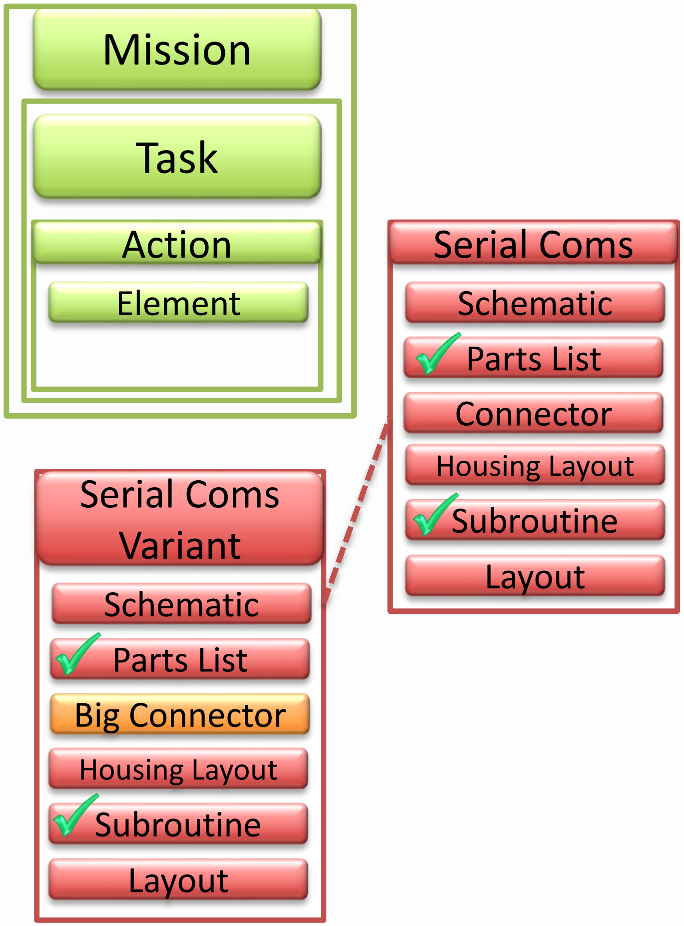
Troubleshooting Problems

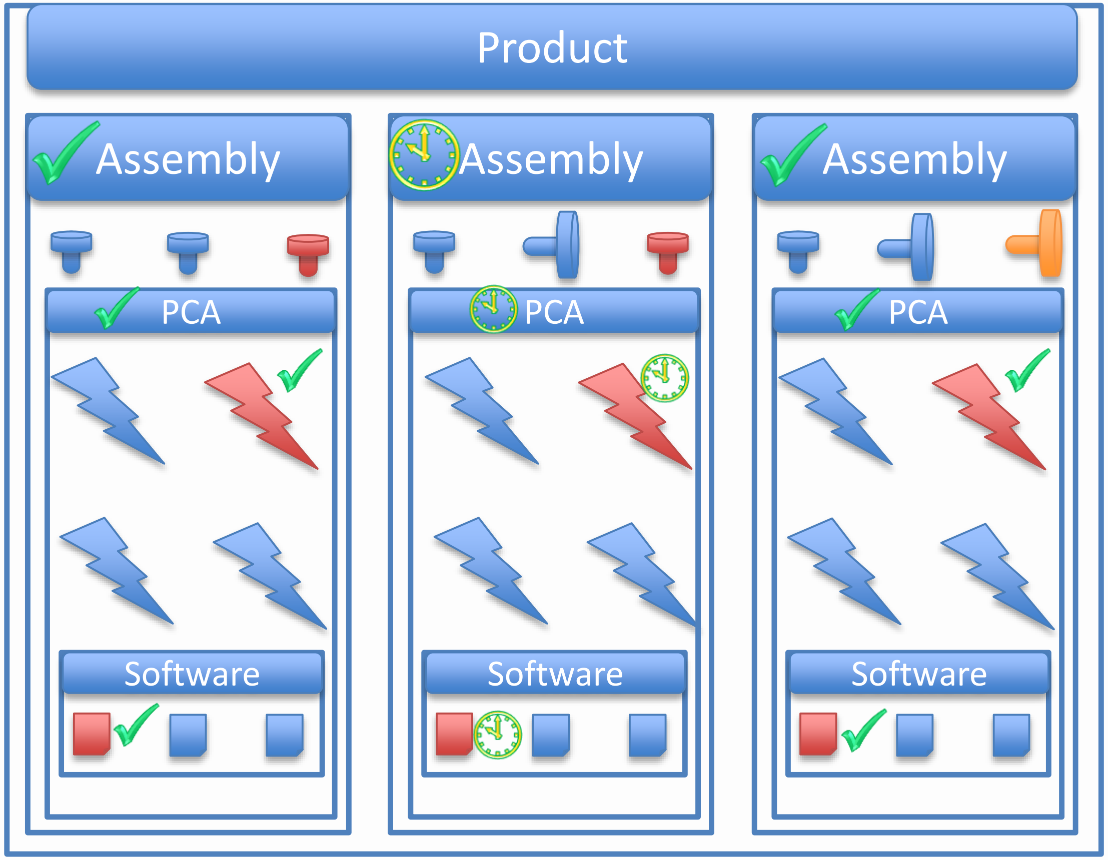


Instant Traceability



Queued Revisions





Agenda

- ✓ Context, History
- ✓ Block Levels
- ✓ Functional Platform-Based Block Structure
- Benefits:
  - ✓ Design Benefits
  - ✓ Sustaining Benefits
  - **Functional Test Benefits**
- Updates

Functional Test Problems

- Functional platform costs

Functional Test Problems

- Functional platform costs
- Fixture development and customization time

Functional Test Problems

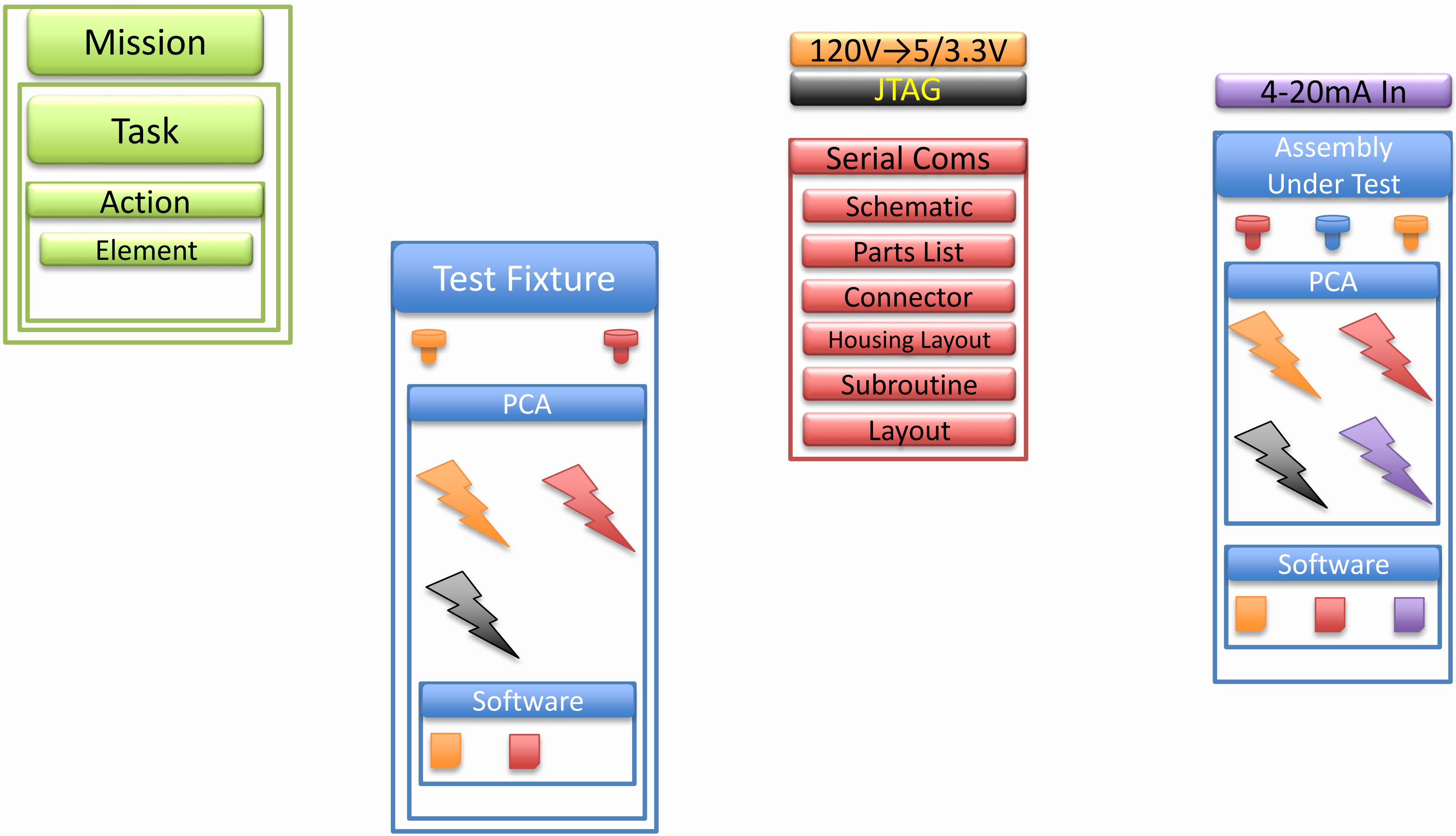
- Functional platform costs
- Fixture development and customization time
- Variations between fixture and product

Functional Test Problems

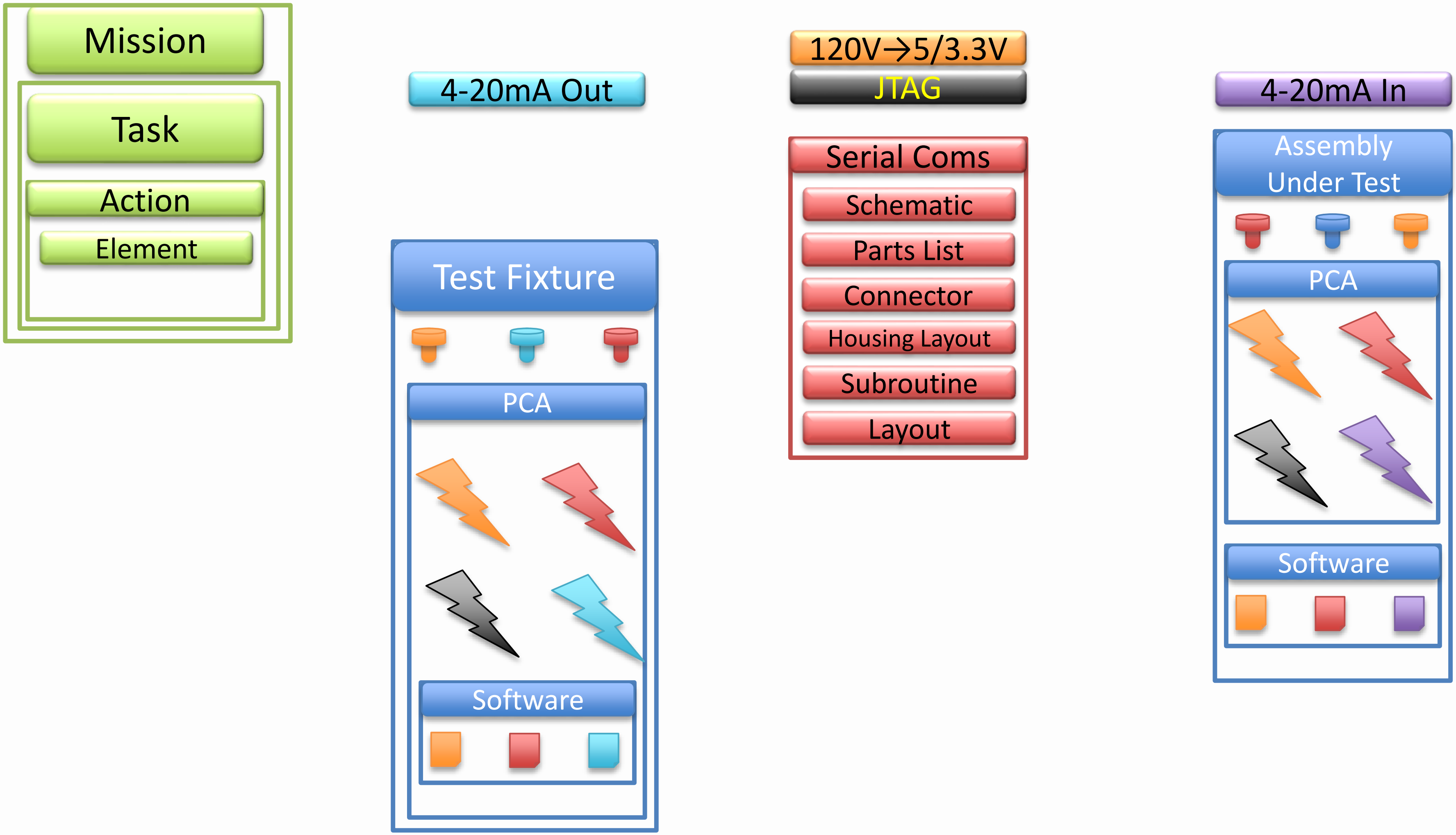
- Functional platform costs
- Fixture development and customization time
- Variations between fixture and product
- Test fixture obsolescence

Common Fixtures and Product

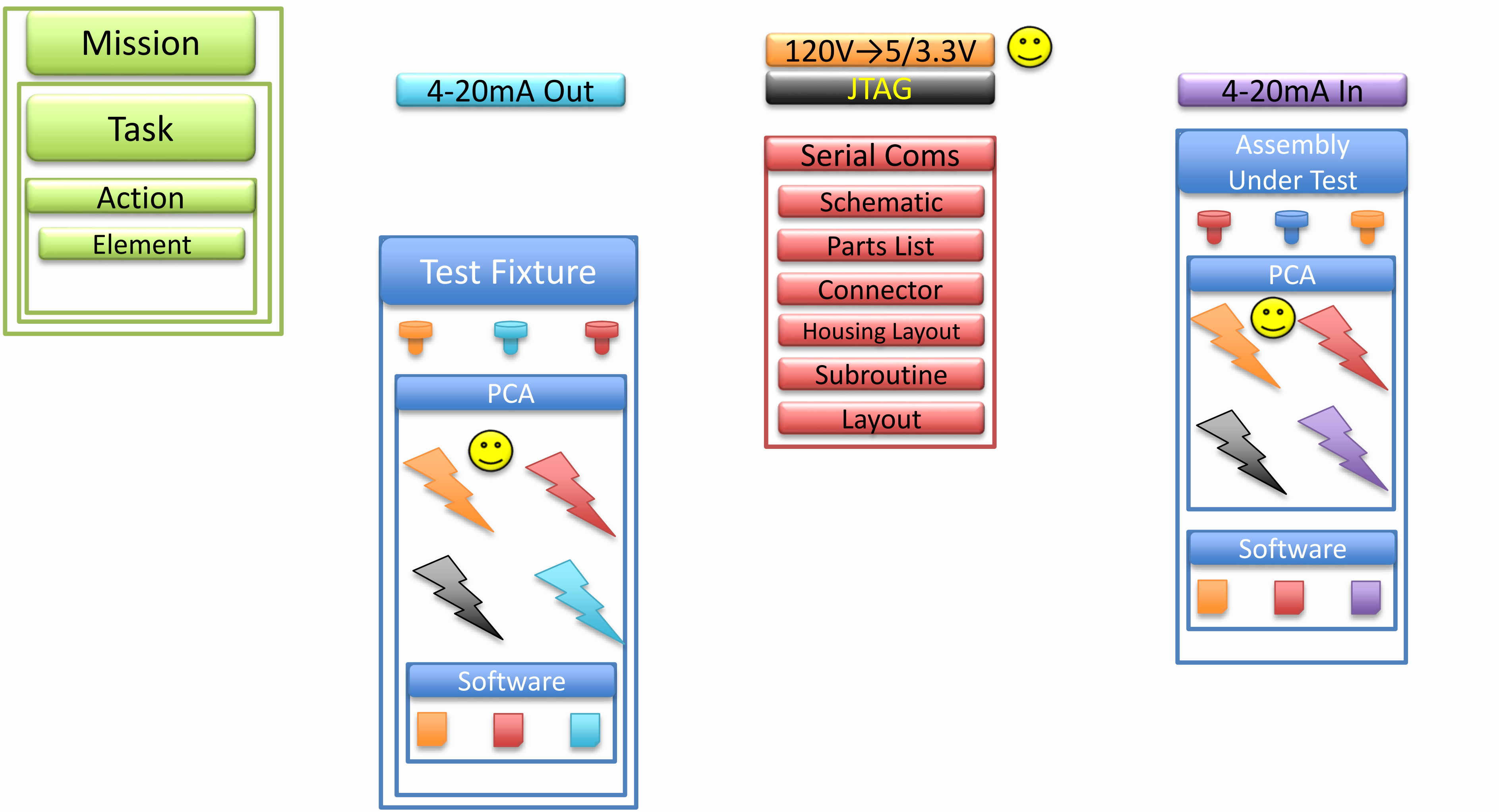
Common Fixtures and Product



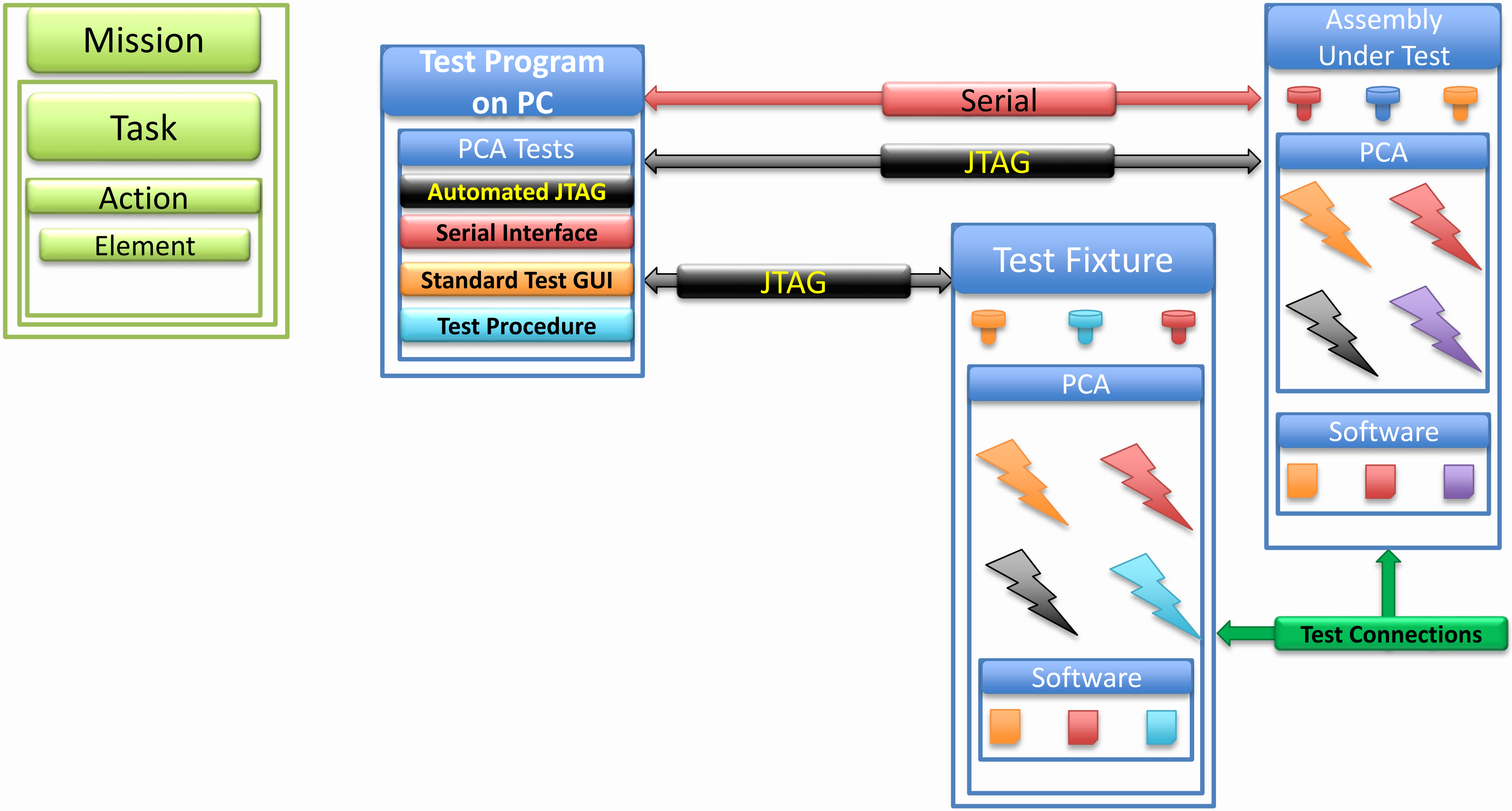
Common Fixtures and Product



Sustaining Engineering Benefits Shared



Test Fixtures with Product Blocks



Summary

- Common Functional Blocks Applied:
  - Across designs
  - Across disciplines
  - To both products and fixtures
- A platform that provides for:
  - Variants
  - Versions
  - Traceability
- Benefits:
  - Save time
  - Improve quality
  - Find and fix errors quickly
  - Focus on function

Agenda

- ✓ Context, History
- ✓ Block Levels
- ✓ Functional Platform-Based Block Structure
- ✓ Benefits:
  - ✓ Design Benefits
  - ✓ Sustaining Benefits
  - ✓ Functional Test Benefits
- **Updates**

Since submission…

APEX EXPO IPC

NEW IDEAS ... FOR NEW HORIZONS

MARCH 25-27, 2014
MANDALAY BAY RESORT AND CONVENTION CENTER
LAS VEGAS, NEVADA

Stephen Golemme
FMC Technologies, Schilling Robotics
Stephen.Golemme@fmcti.com