5. Computing and measurement bus – At the center of the common *core* is a computer in the form of a desktop PC, server workstation, laptop, or embedded computer as used with PCI extensions for instrumentation (PXI). An important aspect of the computing platform is the ability to connect (and communicate) to the wide variety of instruments in a test system. There are several different instrumentation buses available for standalone and modular instruments including GPIB, USB, LAN/LXI, PCI/PXI, and PCI/PXI Express. These buses have differing strengths, making some more suitable for certain applications than others. For example, GPIB has the widest adoption for instrument control and wide availability of instrumentation; USB provides wide availability, easy connectivity, and high throughput; LAN is well-suited for distributed systems; and PCI Express provides the highest data throughput performance.

2.2. CTP – Functional Overview

To better understand the CTP concept, an analogy can be made to a master-planned community where a "strategic" approach is applied to the entire community's needs so that the common infrastructure of the community is designed and developed before the specific structures of the community are established. In likeness to this model for design, the *core* of the common tester is in comparison the infrastructure of the master-planned community (telecommunications, transportation networks, utilities, recreational, business, etc.) while the specific structures built on this infrastructure would be the specific tester applications within the tester community. Just as the common infrastructure would be established for a master-planned community, the infrastructure of the common tester (user interface, instrument drivers, requirements tracking, report generation, soft front panel templates, data management, flow control, etc.) would be the common foundation on which all tester applications would be constructed.

The CTP architecture must be capable of supporting all functionality common to all weapons systems across multiple existing and future weapons programs. These functionalities include but are not limited to the following:

1. Operational support of an accelerated schedule (laying tracks in front of an oncoming train). This refers to the reduced timeline requirement. To meet this requirement, each stage of the tester must be a subset of the next. The CTP ensures the sharing of common tasks, instruments, functionalities, etc., across multiple testers.

2. Grow with the program. As a weapons program moves along the development path, the tester must be capable of testing at the current development level when that level is ready to be tested. For example, the development levels for a radio detection and ranging (RADAR) program are defined as die, multi-chip-module (MCM), subsystem transmit or receive (Tx or Rx), and full radar. Each level has feature specific tests and tests common to other levels. The CTP will ensure that duplication of tests and functionalities is minimized and that functional test and device reuse is maximized.

3. Handle data. The tester must be capable of collecting data, processing at the test station, and storing for later in-depth analysis and data mining. Ideally, data could be collected throughout development through production for comparison over the lifetime of the product.

4. Combine and automate test sequencing. It will be possible to build a test by manually controlling the various instruments necessary to perform that test and then capture the test steps into a sequence for automatic operation. Further, because of the CTP it will be possible to load a sequence from one location into the tester at another location.

5. Be adaptable to add new measurement compatibilities. To keep future development costs low, the common *core* will easily accommodate new technologies and configurations with seamless integration of testing capabilities, growing the CTP and all spawned testers.

6. Meet advanced technology developmental testing requirements. As new technologies become available, they will be incorporated into various weapon systems. These new technologies will require cutting-edge functionalities that have not yet been applied to tester systems in the nuclear weapons complex. By adding these functionalities to the common *core*, the CTP and all spawned testers will transparently upgrade to embrace these functionalities.

To meet the constraints listed above the common *core* needs to be instrument agnostic. In other words, the *core* should not be built specifically to the requirements of one or more instruments. Instead, the software should provide a means of adaptation to any instrument on any bus. To achieve this goal, the software needs to be modular in its design and approach so that it can be adapted across multiple stages of development and weapons systems (Figure 6).

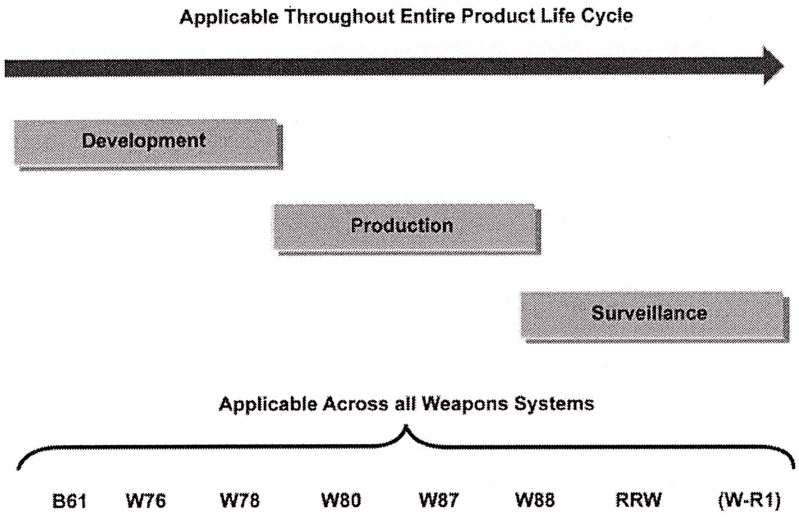
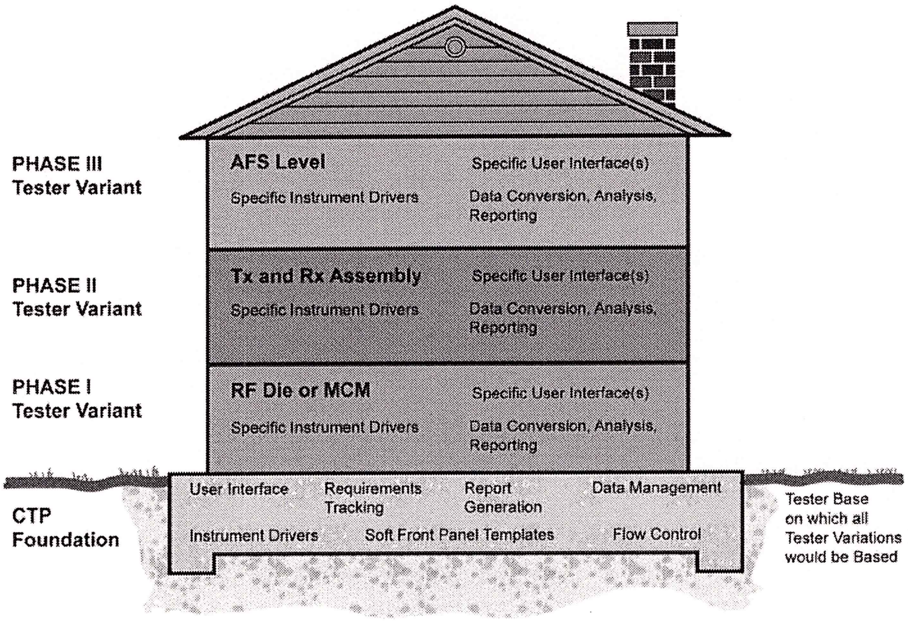

Applicable Throughout Entire Product Life Cycle
Development
Production
Surveillance
Applicable Across all Weapons Systems
B61
W76
W78
W80
W87
W88
RRW
(W-R1)

Figure 6. Application of a Common Platform Overview

The CTP *core* infrastructure must be capable of supporting all tester functionality common to all weapons systems across existing and future weapons programs. Figure 7 shows how a CTP would be applied on a RADAR program. In this figure, an initial *core* tester platform is developed as the foundation for the building shown in the illustration. After establishment of a solid foundation capable of supporting the successive levels, construction can begin on the successive floors of the building for the specific tester applications in each phase of the project. The CTP is applied at the successive phases of the project (the floors in the illustration RF Die and MCM level, etc.) and customized for the needs of that development phase. It is expected that at each level, there will be specific software and hardware development tasks that may become part of the growing code reuse library. This reuse library would include the drivers developed for instrumentation that can be reapplied as it is reused at each level in the structure.



PHASE III
Tester Variant
AFS Level
Specific User Interface(s)
Specific Instrument Drivers
Data Conversion, Analysis, Reporting
PHASE II
Tester Variant
Tx and Rx Assembly
Specific User Interface(s)
Specific Instrument Drivers
Data Conversion, Analysis, Reporting
PHASE I
Tester Variant
RF Die or MCM
Specific User Interface(s)
Specific Instrument Drivers
Data Conversion, Analysis, Reporting
User Interface
Requirements Tracking
Report Generation
Data Management
Instrument Drivers
Soft Front Panel Templates
Flow Control
CTP
Foundation
Tester Base on which all Tester Variations would be Based

Figure 7. CTP Structure Using a *Core* Foundation

Each *core* component of the CTP foundation is discussed in detail below:

1. User Interface – The foundational User Interface will consist of a developmental environment allowing for complete control and ultimate flexibility of the System Management Software. This user interface will then require some customization to support test engineers at each development level. This will lead to specific user interfaces with defined functionality appropriate to the level of expertise of the end users responsible for each level.

2. Instrument Drivers – A body of instrument drivers will be built to support communication to the wide variety of instruments that will be used in various aspects of

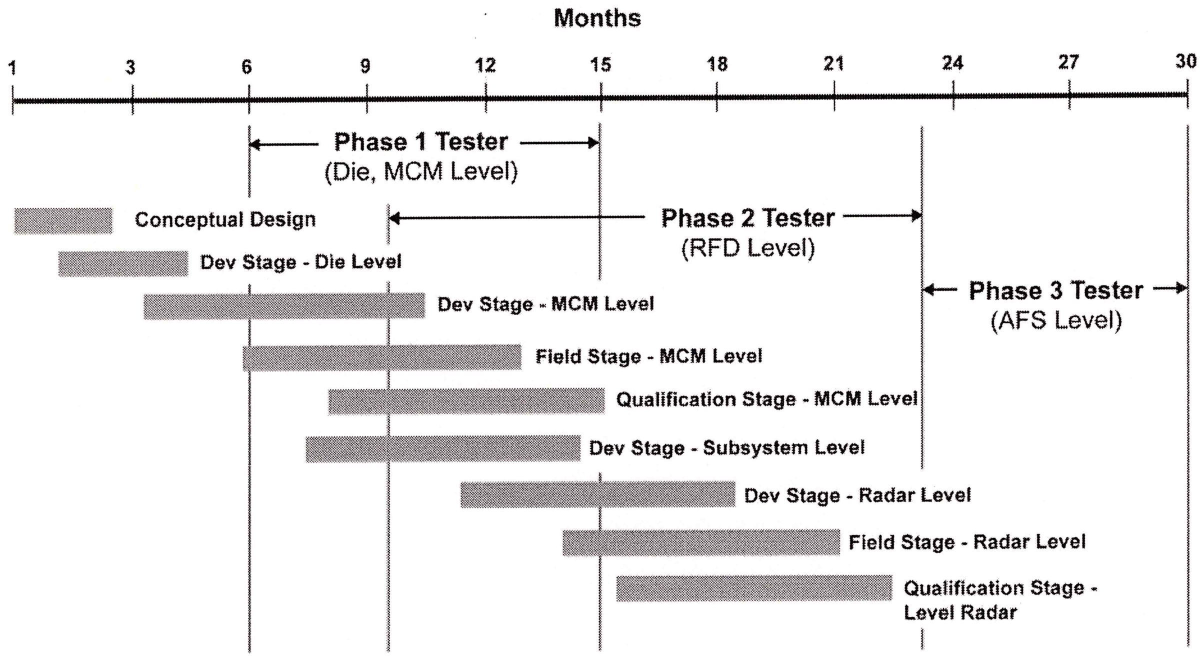


Figure 10. AFS Development Phased Approach to Tester Design

For the AFS program, the first phase of testing involves RF Die level testing. This type of testing requires the deployment of the tester with a microprobe station to support evaluation of RFIC prototype designs. As the RFIC designs mature, the test platform will mature and move towards MCM testing. At that point in time, the CTP architecture can be applied in other areas of the complex as well as in support of the MCM level.

Figure 11 is an initial schematic of the overall architecture of the Die level tester. Software development takes considerably more time and is in process at this point. Six instrument drivers have been completed along with integration of these instruments into the CTP TestStand architecture. In addition, a viable Data Management scheme has been defined and implemented for demonstration. From the hardware perspective for Phase I and Phase II, the design is complete with a CTP that supports a variety of instruments over multiple buses. A Software Requirements Specification (SRS) for the CTP is also under development.