Two Thirds of SDR is SD
Tutorial Outline

1. Introduction
2. State of the Domain and State of the Software Art
    a. The State of Software Technologies as of 2010
    b. The SDR Domain
        i. Characteristics
            1. Distributed
            2. Real-time
            3. Embedded
            4. High Performance
            5. Fault Tolerant
            6. Dynamic
            7. Secure
            8. Networked
            9. Heterogeneous
                a. With regards to Networking
                b. With regards to Software Technologies and Languages
            10. Portable
            11. Standards Compliant
            12. Life Critical
            13. Multi-threaded
            14. Multi-process
            15. Object Oriented
            16. C/C++
            17. Software Defined
            18. Component Based
            19. Resource Constrained
            20. Mobile
        ii. Complexities
        iii. Degrees of Freedom and Axes of Change
            1. Hardware (x86, arm, ppc and various versions thereof)
            2. Operating System (Linux, Integrity, VxWorks ...)
            3. Middleware (vendors, configurations)
            4. Core Framework (vendors, versions)
            5. Optimization Levels (space - speed)
            6. Memory Footprint
            7. Processor Power (size weight and power)
            8. Virtualization
            9. Transports (TCP/IP, Shared Memory, Custom IPC)
            10. Tool chains
            11. Kernel Versions
        iv. Unique Aspects
        v. Non Unique Aspects

vi. The Software Aspect and the Radio Aspect
    1. Intersection
    2. Union
vii. Complexity Ceilings

c. Learning from other domains
d. Two Thirds of SDR is SD
    i. What does Software Defined mean really?
e. Concentrating on the hard problems of the domain … if you can

3. Making the Technologies Fit the Architecture
    a. The Power and Art of Abstraction
        i. A Comment on code duplication
    b. The Difficulties and Realities of Refinement
4. Recent critical software innovations and how they can be applied to the SDR domain
    a. Deployment and Configuration Frameworks
    b. Model Driven Engineering and Domain Specific Languages
        i. Domain Specific Languages
        ii. Rich Domain Specific Editors and Views
        iii. Domain Specific Generators
        iv. Constraint Technologies
        v. Transformation Technologies
    c. Agile Software Development
    d. Automated Test
    e. Continuous Integration
    f. Virtualization
    g. Software Product Lines and Product Line Architectures
    h. Combining and Tailoring all of the above for the SDR domain
    i. Stovepipe breaking technologies
5. Lessons Learned from the Last Decade of Making SDRs
    a. Enabling technologies and solving the problems they introduce
    b. How to develop these systems quickly
    c. The whole of the technologies working holistically is much greater than the sum of the benefits of the individual parts