BeSSt Working Group Goals

• **Motivation**
  
  - Requirements define and scope our systems. Latent defects in requirements are the most expensive to fix.
    - They therefore represent our greatest cost savings opportunities.
  
  - Requirements engineering practices remain relatively stagnant.
  
  - What we teach in school about requirements is rarely reflected in practice.

• **Can we do better than “shall statements” in requirements engineering?**
• Sample Topics for Discussion

- How can we move **beyond English-language sentences** to capture requirements?
- How can we **harmonize our approaches** to requirements engineering with modern techniques for architecture, implementation, and testing?
- Should we continue to **separate requirements engineering** from design, or try to integrate them more closely?
- Should we adopt **agile or fluid methodologies**, where requirements evolve along with the system, rather than being developed all-at-once up front? What are the implications for development and contracting models?
- How should we **train the next generation** of engineers?
- How many requirements is **too many, or too few**?
BeSSSt Invited Presenters/Panelists

- **Professor Richard N. Taylor, UC Irvine**
  - Classical requirements engineering practice has failed to deliver. A way forward exists, based upon software architecture.

- **Ban Al-Ani, UC Irvine**
  - Current requirements engineering approaches are inherently hubris: they do not take into account the target users’ context and environment.

- **Jorge Seidel, Aerospace Corporation**
  - Why do we use fixed, document-based requirements in a world of hyper-exponential change?
BeSSSt Invited Presenters/Panelists

- Andrea Richards, Raytheon
  - Why do we lack a strong feedback loop from test and verification back to requirements development for the next generation of systems?

- Dale Robinson, Raytheon
  - The number of requirements should be as few as possible: this allows for a wider trade space.

- Emil White, Lockheed-Martin
  - Why do we continue to develop requirements without regard for how they will be verified, when this causes trouble for us all the time?