THE SCIENCE OF SYSTEMS ENGINEERING

George A. Hazelrigg
Deputy Division Director
National Science Foundation
Division of Civil, Mechanical and Manufacturing Innovation

Conventional wisdom

- Systems engineering is a process
- Decision making approaches are ad hoc
- If it doesn't work, it's because you didn't adhere to the process
- The goal is to get a good (near optimal) system design

SE implementation

- Develop system-level requirements
- Flow down requirements to subsystem levels
- Analyze/design at the subsystem level in a deterministic environment
- Use ad hoc approaches to deal with risk

Requirements

- Requirements are constraints, they only specify what is not wanted
- Requirements do not represent preferences and thus do not enable system optimization
- As constraints, requirements impede the development of better system designs
- The process of requirements flowdown reduces the likelihood that system-level requirements will be met

Treatment of uncertainty

- We fail to teach predictive modeling
- Commonly used methods can be grossly in error
- Methods commonly used for decision making under uncertainty are ad hoc
- There is provably only one correct approach, probability-based utility theory

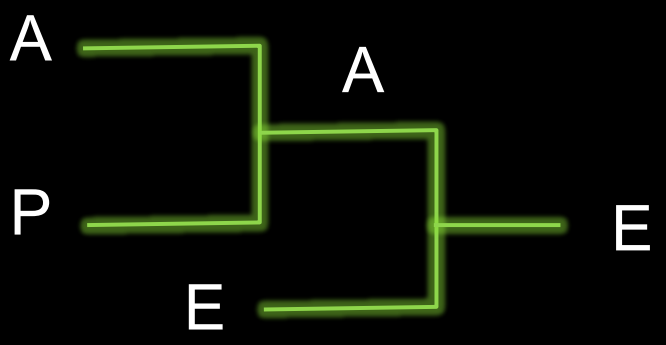
Group processes

- Groups do not make decisions—only individuals make decisions
- Groups have emergent behaviors, typically path dependent
- Failure to recognize the problems can be catastrophic
- The mathematics of group processes is called game theory

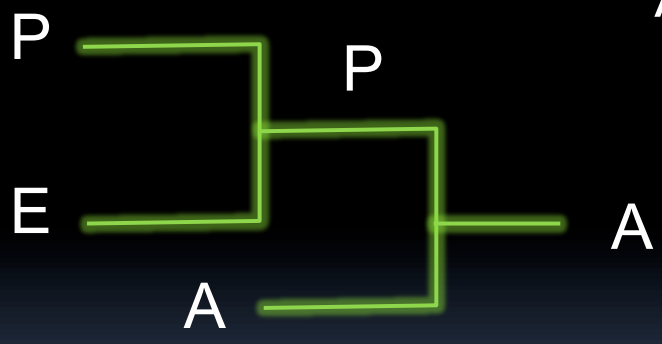
Example

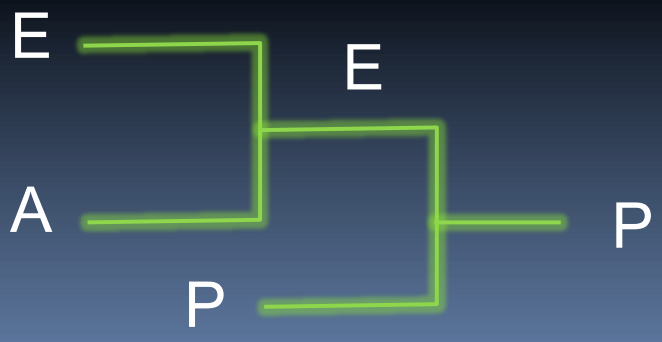
- Three people select a pizza pie to order (A-anchovy, P-pepperoni, E-extra cheese):
  - Juan A > P > E, A > E
  - Pedro P > E > A, P > A
  - Maria E > A > P, E > P
- The group preference is A > P > E > A
- Regardless which pie is chosen, there is a better one—there is no optimum pie

Three ways to choose a pie



- Juan A > P > E, A > E
- Pedro P > E > A, P > A
- Maria E > A > P, E > P
- The group preference is A > P > E > A





The result depends on the path that we take to get to there

Continuous improvement

- We have product S, can we improve it?
- Suggested changes A, B, C
- Possible products: S, S_A, S_B, S_C, S_{AB}, S_{AC}, S_{BC}, S_{ABC}
- Product improvement team:
  - Jan $S_{AB} > S_A > S > S_B > S_C > S_{AC} > S_{BC} > S_{ABC}$
  - Pat $S_A > S > S_B > S_C > S_{BC} > S_{AC} > S_{ABC} > S_{AB}$
  - Michael $S_B > S > S_C > S_{CB} > S_{AC} > S_{ABC} > S_{AB} > S_A$

Continuous improvement

- Given the team's preferences:
  - Jan $S_{AB} > S_A > S > S_B > S_C > S_{AC} > S_{BC} > S_{ABC}$
  - Pat $S_A > S > S_B > S_C > S_{BC} > S_{AC} > S_{ABC} > S_{AB}$
  - Michael $S_B > S > S_C > S_{CB} > S_{AC} > S_{ABC} > S_{AB} > S_A$
- $S < S_A < S_{AB} < S_{ABC}$
  - The team thus moves from S to S_A to S_{AB} to S_{ABC}
  - But everyone agrees that $S > S_{ABC}$
- Continuous improvement made the product worse

The bottom line

- Ad hoc methods are hurting us big time
- We need to understand that systems engineering is a group process within which individuals are making decisions
- We need to understand what can go wrong
- We need rigorous theory to enable damage control
- Rigorous theory can be built on probability theory, decision theory and game theory