Nancy Leveson:
Software Engineering: A Look Back and A Path to the Future

- Purpose of the article
  - The engineering of software systems
  - Differences with other engineering disciplines. What are they?

- Achievements in the field. What are they?
  - Engineering discipline into the software development process.
  - Role of abstraction and separation of concerns, modularity, lifecycles, processes, measurements and metrics
  - Can these be adopted directly from other engineering disciplines? Why/why not?
    - Software is infinitely flexible and non-physical in nature
Other achievements?
- application of mathematical rigor to software development
  - Formal specifications
  - Formal verification

How has the early emphasis on efficiency changed? To what has it changed? Why is this the case?
- More complex systems come with their own set of problems
  - Teaming, management, and communications
- What used to be a problem of developing new software has turned into a problem of maintaining and enhancing legacy ones.

It is a problem of Intellectual manageability and human limits.

What does the author suggest as solutions?
- Cognitive psychology and engineering
- Software as a product that assists humans
- Combination of problem-solving techniques to arrive at products useful and usable by those humans
David Parnas:
Software Aspects of Strategic Defense Systems

Collection of eight essays describing why the software systems required by the Strategic Defense Initiative Organization (SDIO) can not be trustworthy.

1. why software is not reliable
2. Properties of SDI software are unattainable
3. Why are techniques to build military software are inadequate
4. Nature of software engineering research is insufficient
5. How would research and AI help build reliable software
6. How would automatic programming bring substantial improvements
7. Why formal verifications cannot produce desired systems
8. Why is funding insufficient
Essay 1: why software is not reliable

- Continuous, discrete and hybrid systems
  - Huge amount of states
  - No similar/repetitive structure to ease testing and analysis
  - Lack of regularity in software systems

Essay 2: why are Properties of SDI software are unattainable

In three groups.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
Essay 3: Why are techniques to build military software are inadequate

- Current method of programming
- Issues with this method
  - Concurrency and multiprocessing amplify these issues greatly

Essay 7: Why formal verifications cannot produce desired systems
David Parnas:
Software Aspects of Strategic Defense Systems

- Essay 8: Why is military funding insufficient