

Software Engineering: Theory and Practice

Conclusion

Tom Verhoeff

Eindhoven University of Technology
Department of Mathematics & Computer Science
Software Engineering & Technology

Feedback to T.Verhoeff@TUE.NL

Future Developments

- Increasing software/system complexity
- Further integration with hardware: sensors, robotics, 3D output
- Further standardization; certification of product, process, staff
- Higher-level formal models, e.g. for architecture, to generate code
- Dynamically evolving, self-adjusting, “organic” code
- Other kinds of processors: neural networks

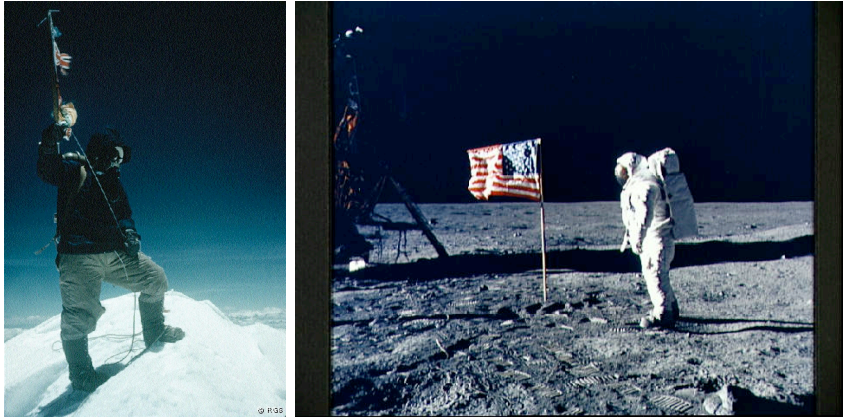
Lasting Principles

- Rigor & formality
- Separation of concerns
- Modularity
- Abstraction
- Anticipation of change
- Generality
- Incrementality

Looking Back

- Software (and) Engineering
- Product, Process, Project, Documentation
- Requirements Engineering
- Configuration Management
- Dealing with Errors in Engineering
- Verification by Review and by Testing
- Architecture
- Software products made by **New Mind Applications**

What Was the (Real) Goal?



© 2008, T. Verhoeff @ TUE.NL

5

Software Engineering: Conclusion

Look beyond Planting the Flag

- Planting the flag is highly visible, but not the ultimate goal.
Though the media would often like you to believe it is.
- Planting the flag brings you halfway, at best.
- The cost after planting the flag exceeds that of before.
- You must prepare for the second half before you depart.
- In software, planting the flag is the yell "it works".
After that comes maintenance (and usually much more).

© 2008, T. Verhoeff @ TUE.NL

6

Software Engineering: Conclusion

References

- Chapter 1 of *The Project Manager's Guide to Software Engineering's Best Practices* by Mark J. Christensen and Richard H. Thayer. Wiley, 2002.

© 2008, T. Verhoeff @ TUE.NL

7

Software Engineering: Conclusion