

Software Evolution: the Laws

Software Evolution – L1T2

Dr. Vadim Zaytsev aka @grammarware, February 2021



Software System Types: S

- S-type programs
 - “specifiable”
 - problem formally defined by a spec
 - automated acceptance possible
 - such software **does not evolve**

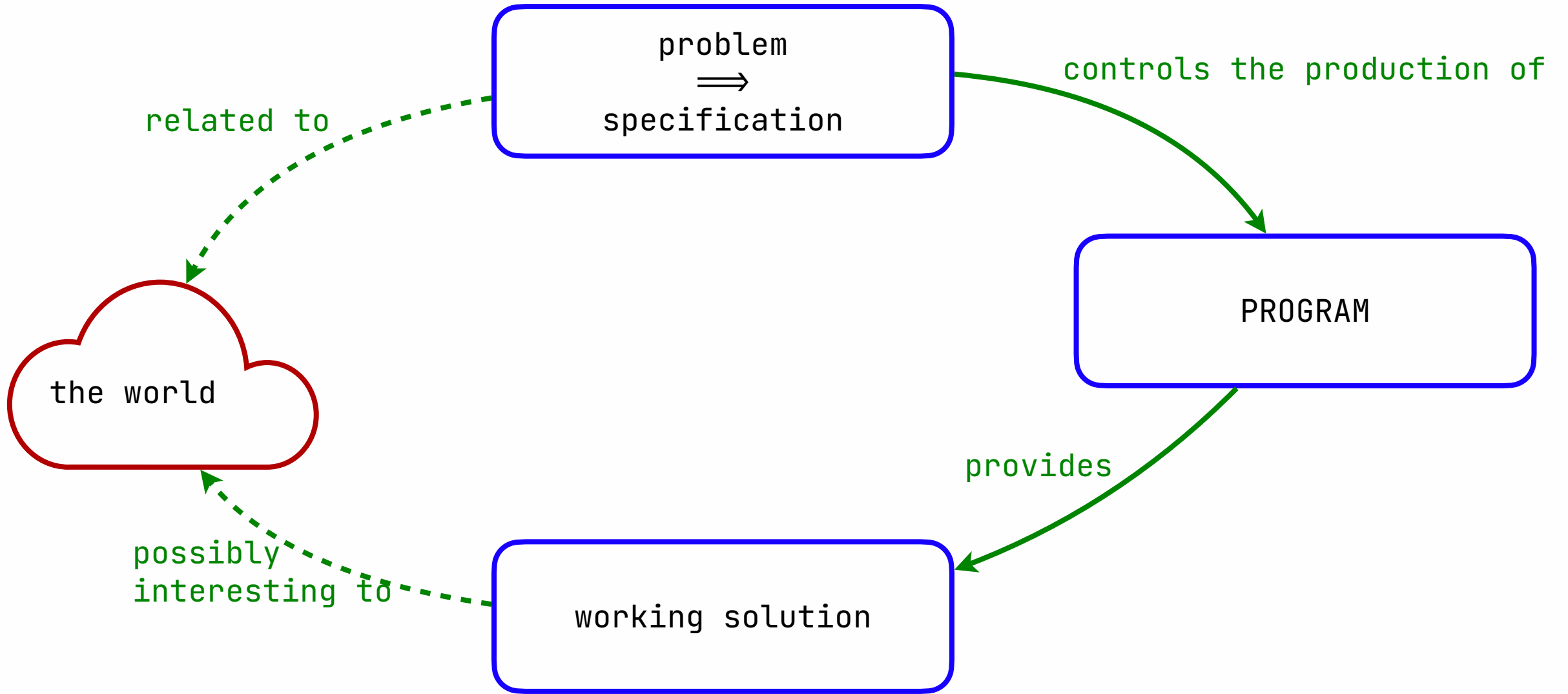


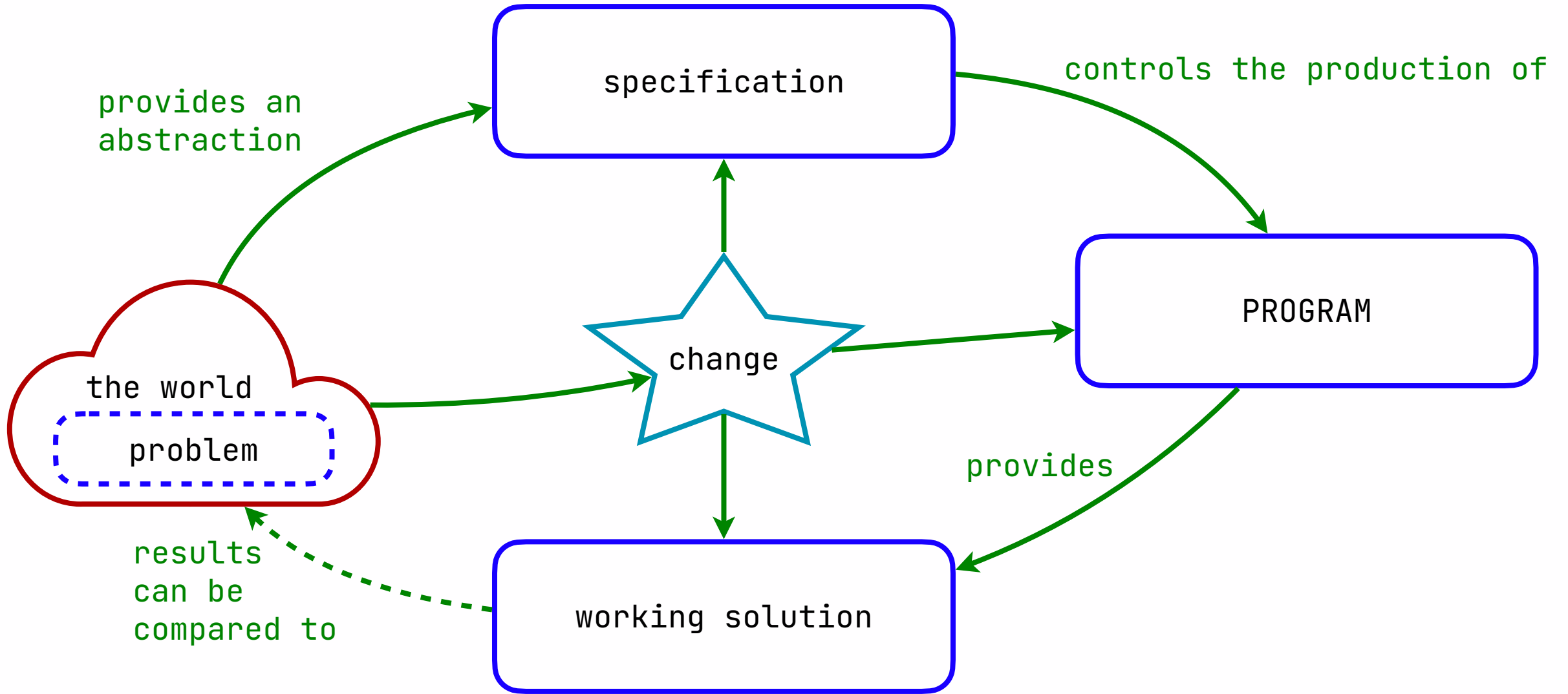
Software System Types: P

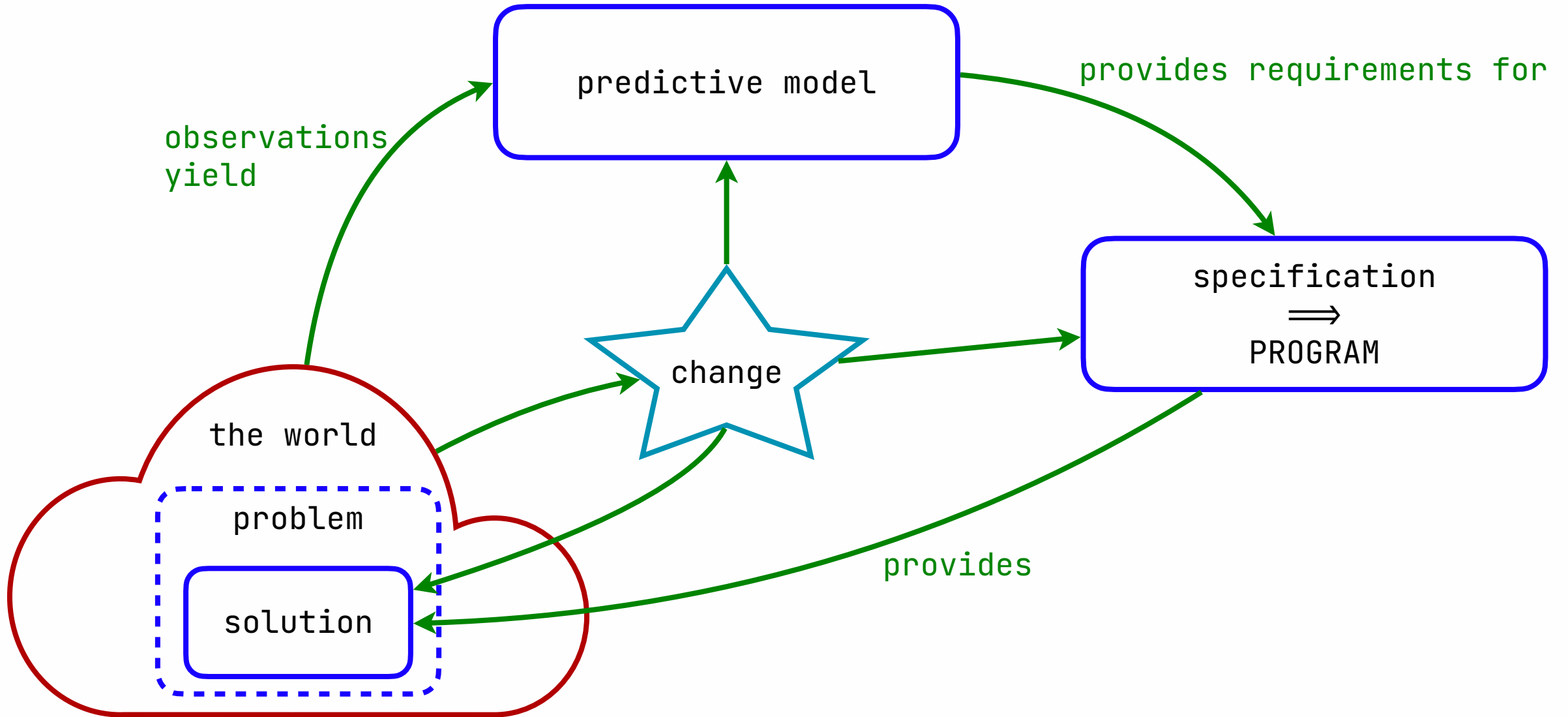
- P-type programs
 - “problem-solving”
 - problem models a real-world task
 - imperfectly
 - qualitative acceptance
 - they can evolve continuously

Software System Types: E

- E-type programs
 - “embedded”
 - solution is a part of the world
 - acceptance is subjective
 - they are **inherently evolutionary**







Lehman's Laws [1/8]

- Continuing Change
 - E-system rots unless adapted
 - the process never stops
 - (true for P-systems as well)



Lehman's Laws [2/8]

- Increasing Complexity
 - E-system becomes more complex
 - evolving means complicating
 - (unless we do something)



Lehman's Laws [3/8]

- Self-regulation
 - E-system evolution is SRP
 - obeys certain statistical laws
 - trends and invariances
 - statistically determinable
 - (distribution close to normal)



Lehman's Laws [4/8]

- Invariant Work Rate
 - E-system dev activity is invariant
 - throughout its lifetime
 - conservation of organisational stability
 - (does not depend on resources)



Lehman's Laws [5/8]

- Invariant Perceived Complexity
 - E-system changes per release
 - statistically invariant
 - throughout its lifetime
 - conservation of familiarity
 - (bored \leftrightarrow overwhelmed)



Lehman's Laws [6/8]

- Continuing Growth
 - E-system must add features over time
 - to keep users satisfied
 - (expectations creep)



Lehman's Laws [7/8]

- Declining Quality
 - E-system perceived quality declines
 - internal as well as external
 - (unless constantly maintained)



Lehman's Laws [8/8]

- Feedback System
 - E-system evolution is a
 - feedback system
 - multi-level
 - multi-loop
 - multi-agent



Lehman's Laws:

- Continuing Change
- Increasing Complexity
- Self-regulation
- Conservation of Organisational Stability
- Conservation of Familiarity
- Continuing Growth
- Declining Quality
- Feedback System



Conclusion

- Software evolves
- Some software evolves faster/differently
- Software evolution obeys certain laws
- Software rots in time (quality, complexity...)

- Q&A Sessions @ Canvas
 - ⇒ v.zaytsev@utwente.nl
 - ⇒ <https://discord.gg/n7VQAPNBPD>