Reflections of Software as Craft

R. DuPlain

2020

1. System of Record

State of the system:
code, tests, docs, data, users;
neither you nor specs.

2. Continuous Integration, Delivery

Through CI/CD,
keep software in working state,
not "done" until shipped.

3. Clean Code Discipline

Explore prototypes
then invest in quality
early in project.

4. How-To Solve It

Understand problem,
devise plan, carry it out,
look back and repeat.

5. Software as Craft

Clear goals, feedback loop,
disciplined craft practitioners
reach a state of flow.

6. An Open Letter

What's technical debt?
The result of non-planning.
It does not exist.

7. Communication as Cost

Use cases, specs, docs,
have noise and non-zero cost,
they're not all value.

8. North Star Navigation

Key to reducing
cost of communication:
goals with permanence.

9. Consensus Across Boundaries

Design by contract
down to surface area
that's easy to test.

10. Deadline as Cost

A "five minute task"
in the backlog for three months
has taken three months.

11. Now or Not-Now

With north star in mind
step in the right direction
strictly: now, not now.

12. Placement of Backlog

Backlog not features
but requests before triage;
work in progress: one.

13. Creativity through Ambiguity

You will never know
at once, everything you know
through iteration.

14. Exposition as Means

Artifacts must have
clarity, resulting from
the writing process.

15. Clarity through Constraints

You can deliver
a keynote presentation
in a lightning talk.

16. Craft as Prerequisite

Ambiguity,
now, not now – total nonsense –
form follows finance.

17. Engineering is Trade-Offs

Discipline not lost,
find proxy variables,
enable trade-offs.