Reviews & Metrics for Product Management

⚠️ User studies are a great way to validate that you are creating the right product for your users

⚠️ Verification
  ↳ when a software product meets all the specified requirements

⚠️ Verification vs Validation

Verification	Validation
Ensures you're building the product right	Ensures you're building the right product

⚠️ When a product is validated, the released software product satisfies the client

⚠️ Monitoring
  ↳ Consists of tracking, reviews, and evaluating the product & the process throughout development

⚠ **Metrics**
  ↳ Are a method of measuring something. You can use metrics in development to measure various aspects of the product, the process & the project

⚠ **Feedback**
  ↳ Is information or criticism that is provided to suggest improvements or to affirm that you're on the right track

⚠ A sprint review meeting is a meeting that is held at the end of a sprint
  ↳ It's an opportunity for the development team to demonstrate their product

⚠ Product Owner is responsible for product backlog

⚠ Sprint retrospective meeting is where the team's process is re evaluated & discussed

⚠ **Scrum roles**
  ① Scrum Master   ② Stakeholders
  ③ Users   ④ Dev Team   ⑤ Product Owner

⚠️ Three main events that occur in a sprint review meeting

- → Product demonstration
- → Product Owner Approval
- → stakeholder feedback

⚠️ The product demo only includes tasks that passed the definition of done

- → coded, tested, documented, usable, ready to be released

⚠️ Types of suggestion at the end of the sprint review

- → praise that the product or a feature is on the right track
- → suggested areas for change or improvement
- → questions, new problems, assumptions

⚠️ Sprient review meetings are a great opportunity to

- → demonstrate hard work
- → increase transparency
- → gain valuable feedback

⚠️ A user study can measure the usability of a product

⚠ Usability

   ↳ the measure of "effectiveness", "efficiency", and "satisfaction" with which specified users can achieve goals in particular environments

⚠ A common way that user studies are conducted is by having a test participant use the product while being observed & recorded by experts

⚠ Goal Driven User study

   → when a user has a specific goal to meet, a user study validates or whether not the product is allowing a user to achieve the goal.

   ↳ It also reveals the parts of the application which may be hindering the user from achieving the goal

⚠ Free form user study

   ↳ where a user is asked to use the product however they feel

⚠️ User shadowing user study

    ↳ where you should shadow your user and see what an average day looks like for the user

⚠️ A/B testing user study

    ↳ where you compare two versions of the product against each other. Generally this is done by releasing two very similar products with one feature change
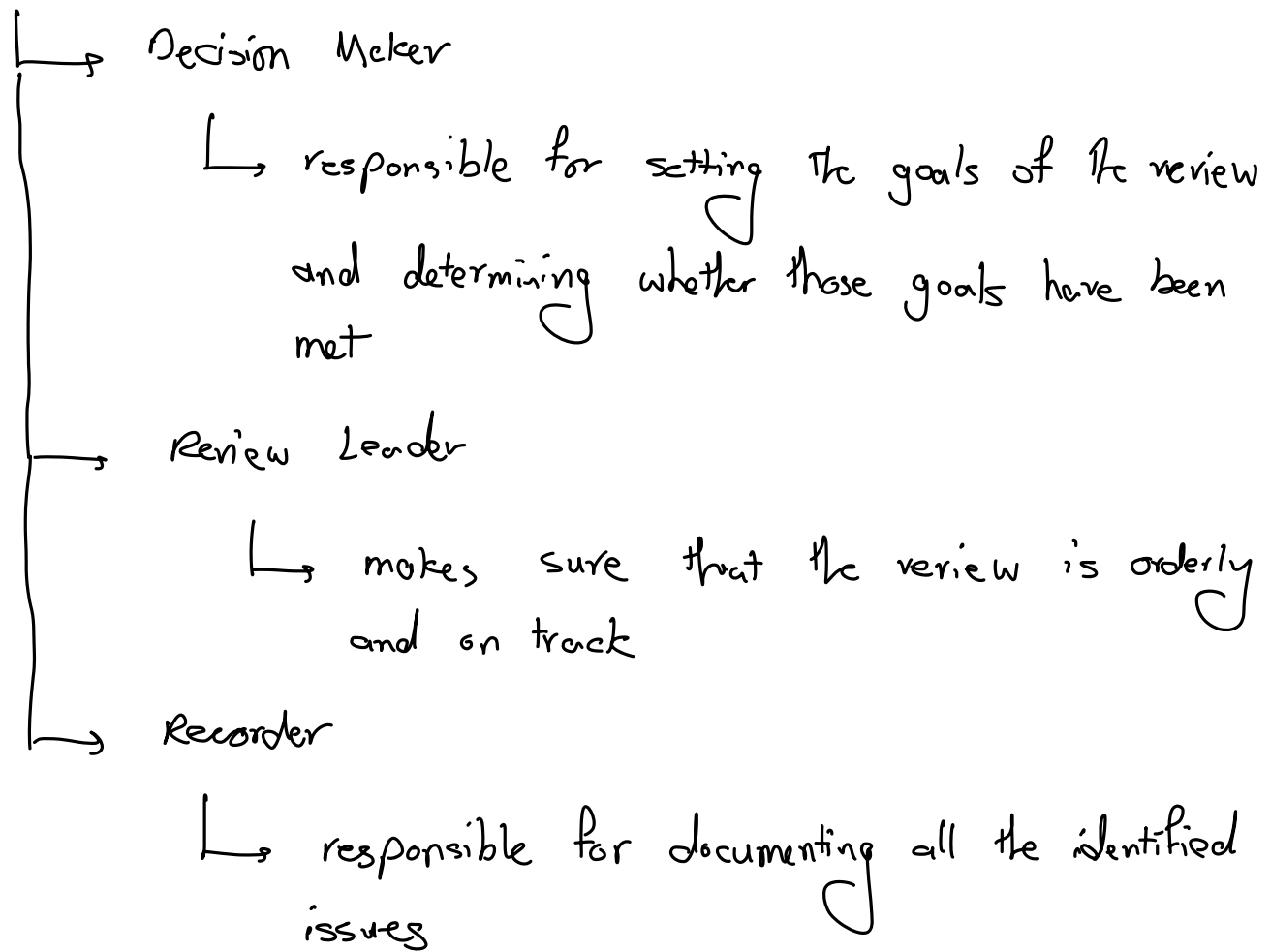
⚠️ 5 stages of design sprint

    ↳ understand, diverge, decide, prototype & validate

⚠️ Design Thinking is focused on identifying people's needs & coming up with solutions to those needs

stages ↳ understand, explore, prototype & evaluate

⚠️ A software technical review could be used in creating more robust designs, forming well defined requirements from a client, or making a number of technical improvements to the code

⚠️ A software technical review is discussion oriented & consists of three main roles
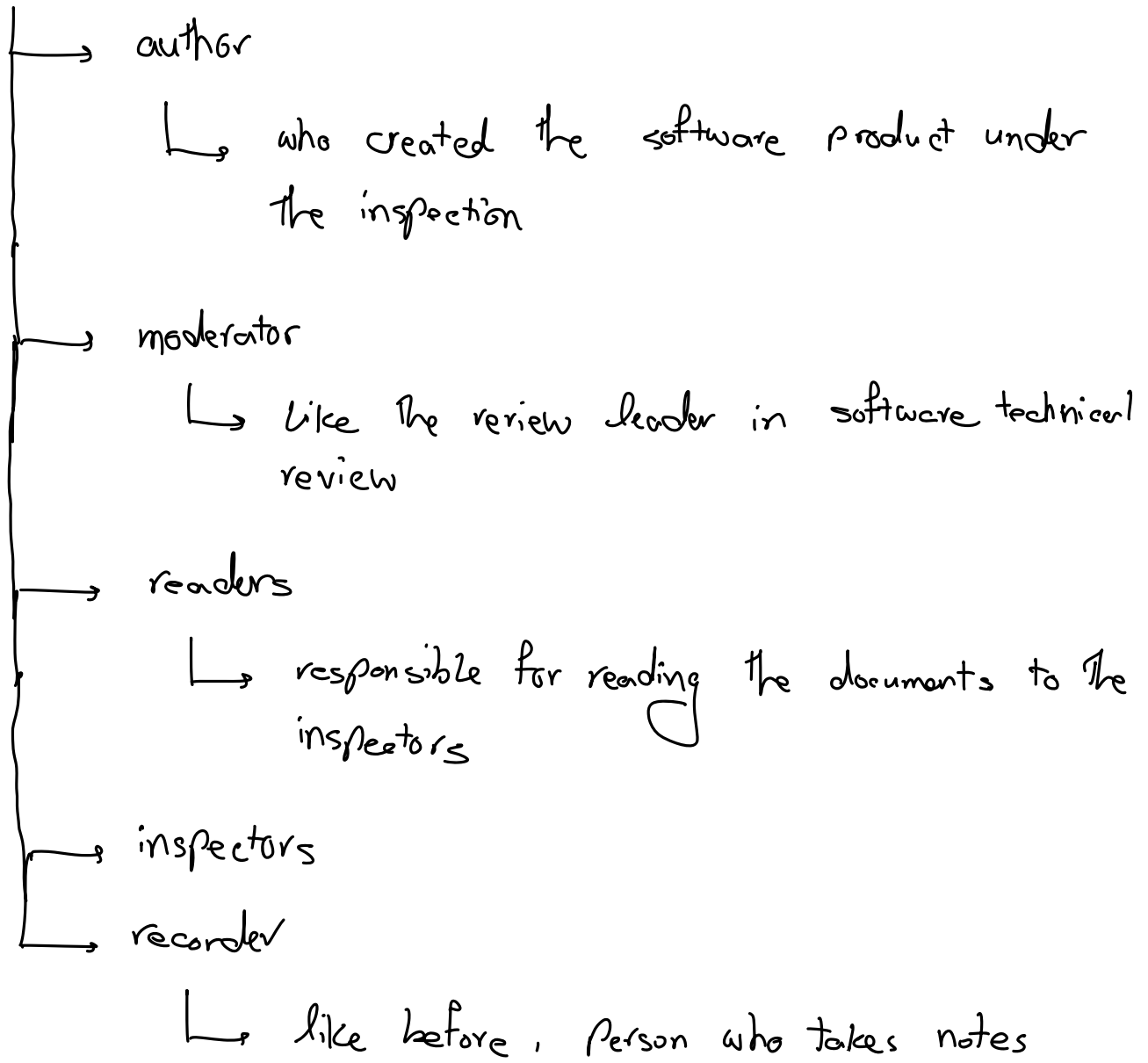
    → Decision Maker

        → responsible for setting the goals of the review and determining whether those goals have been met

    → Review Leader

        → makes sure that the review is orderly and on track

    → Recorder

        → responsible for documenting all the identified issues

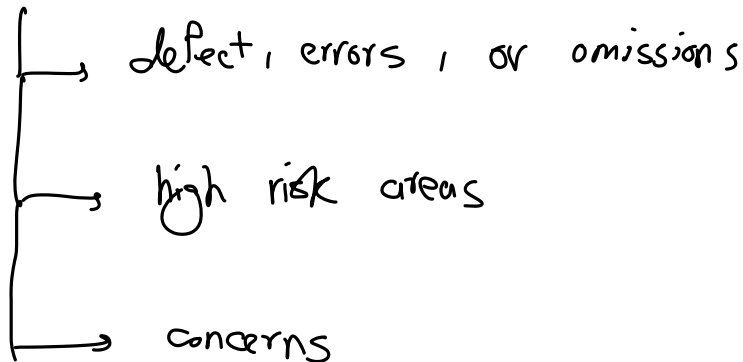⚠️ At the end of software technical review, a certain goal should've been met

⚠️ A software inspection is more formal than previous ones

    → the main purpose is to find and fix defects in a work product, like a set of requirements for a modular code
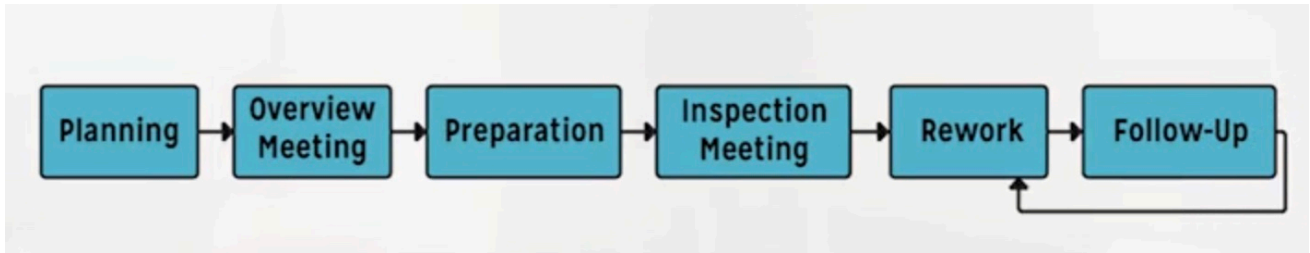
⚠ Software Inspection roles

- → author
  - ↳ who created the software product under the inspection

- → moderator
  - ↳ like the review leader in software technical review

- → readers
  - ↳ responsible for reading the documents to the inspectors

- → inspectors

- ↳ recorder
  - ↳ like before, Person who takes notes

⚠ Requirements technical review output

- ↳ defect, errors, or omissions

- ↳ high risk areas

- ↳ concerns

⚠️ **Software inspection stages**

Planning	→	Overview Meeting	→	Preparation	→	Inspection Meeting	→	Rework	→	Follow-Up

⚠️ lines of code per sprint is a poor metric for assessing software productivity because "lines of code" is a poor measure of the amount of work done

⚠️ Goal, Question, metric (GQM) is a method of choosing metrics for your software projects

⚠️ A metric is a quantitative way in which data can be collective to address the question you're asking

⚠️ Objective Metrics
　　　↳ Based on data that does not rely on anyone's individual viewpoint or interpretation
　　　　　↳ ex : # of product downloads

⚠️ Subjective Metrics
　　　↳ Are those that could depend on interpretation
　　　　　↳ ex: product ratings by users

⚠️ **Measure**
┗→ a standard or unit of measurement. Basically, it's just one instance of a measurement

⚠️ **Metric**
┗→ The combination of two or more measures, that gives a meaningful result

⚠️ **Indicators**
┗→ Measures That draw the attention of the person who is making measurements

⚠️ Metrics should be simple & intuitive

⚠️ Common Categories that are often measured on software projects using metrics
┣→ Maintainability
┣→ Reliability
┗→ Performance

⚠️ Defect Analysis is a way to assess product quality

⚠️ The daily scrum is used to monitor progress
┗→ time boxed : 15 minutes

⚠ Daily Scrum Agenda

- → what did you accomplish yesterday?
- → what will you do today?
- → Are there any impediments in your way?

⚠ Scrum Master is not always responsible for starting the daily session

⚠ Velocity

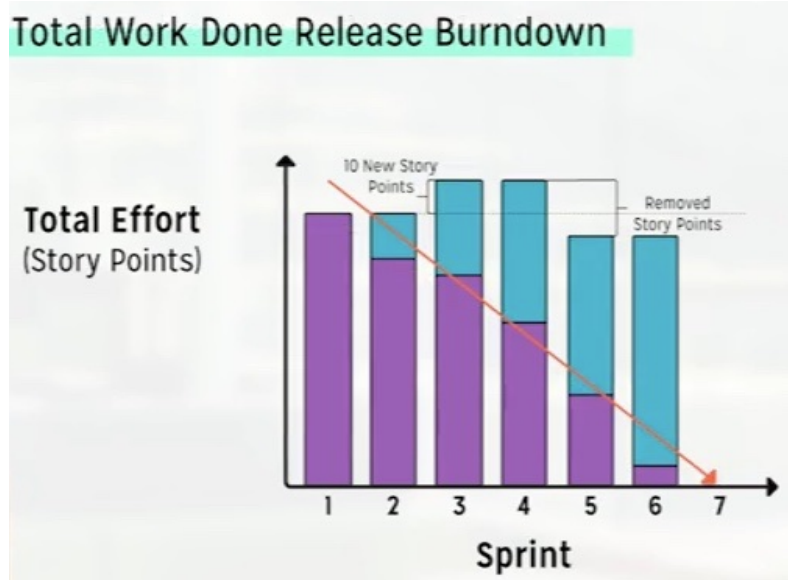- → the measure of the number of units of work a team completed in a given time interval
  - → This is commonly measured in story points completed per sprint
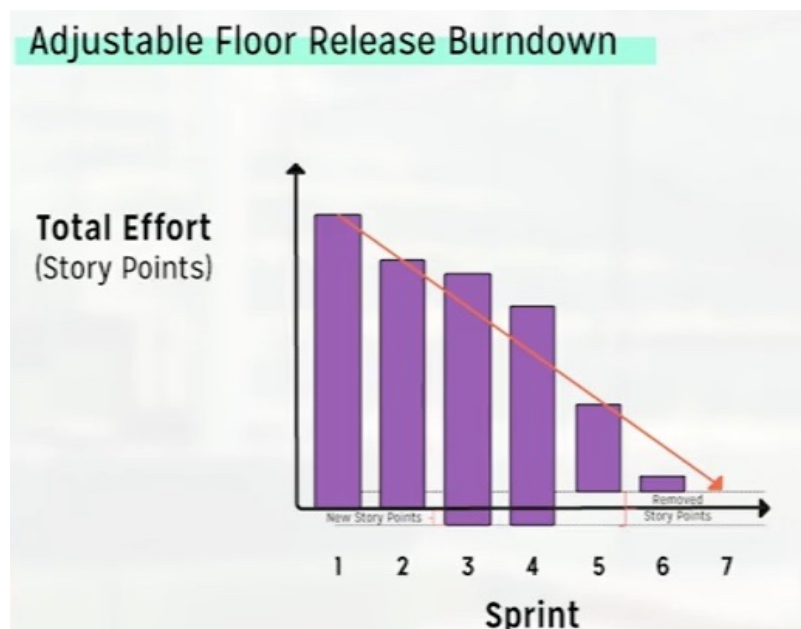
⚠ A release burndown chart

- → will show you how much work the development team has completed, how much is left to complete, The team's velocity in each sprint, and when you should expect to finish the product

⚠️ Two common ways to demonstrate changing requirements in a release burndown chart

↳ ① show the total work done in each sprint above the bars for work remaining

Total Work Done Release Burndown

Total Effort (Story Points)

10 New Story Points

Removed Story Points

1 2 3 4 5 6 7

Sprint

↳ ② by using a adjustable floor

Adjustable Floor Release Burndown

Total Effort (Story Points)

New Story Points

Removed Story Points

1 2 3 4 5 6 7

Sprint

⚠️ In an iteration burndown, we are tracking the completion of tasks whereas in the release burndown, we're tracking the completion of user stories

⚠️ Retrospective

  └─> The act of reflecting back on work that was done and not done in the project

⚠️ Project retrospective ≃ postmortem

⚠️ Sprint retrospective meeting should be the last event That occurs in the sprint

⚠️ Sprint retrospective Questions

  ├─> what went well this sprint
  ├─> what didn't go well this sprint?
  └─> what could we improve for next sprint?