

Shepherding Cats

miguel sicart
game design

today's menu

- design thinking about methods
- agile (software) development
- scrum
- how to do this? practicing scrum
- benefits and pitfalls of scrum
- I am not right - but you should know this



Iterations = Good



as long as you have control ...

development control:

project managers are not evil



Games are business

- Developing games is expensive (even for casual products)
- Developing games is a multidisciplinary, creative job
- One flop can take you out, and ideas are not there everyday

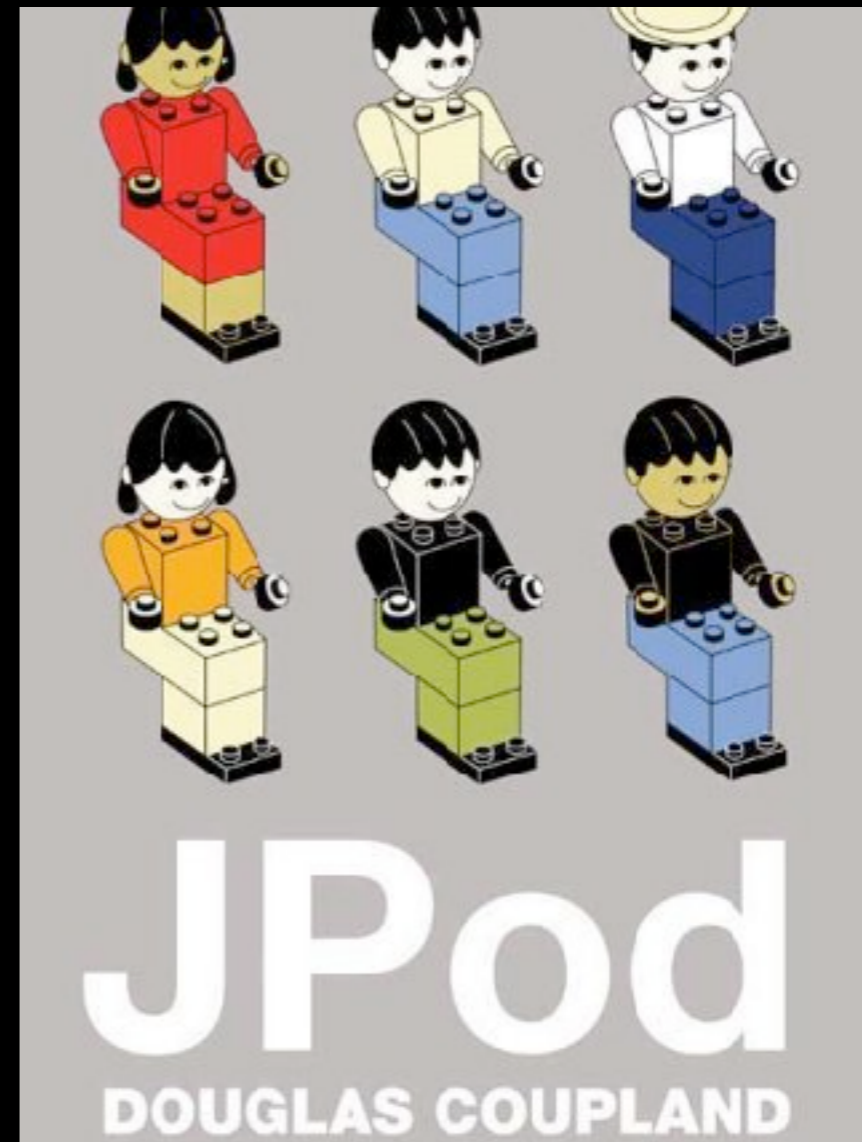


Making games is like shepherding
cats

How to survive?

Development methods

- Work must be coordinated in different phases.
- Goals per phase have to be identified,
- teams must be assigned,
- schedules must be met,
- and budgets are sacred.



Why Agile?

Increasing development costs,
better turnaround times and
production cycles, more
competent middleware tools,
better educational
frameworks, ...



better psychological rewards

less purgatory time



Manifesto for Agile Software Development

- Change is good - also in late stages
- Business side and developer side should cooperate
- Sustainable development
- Steady deliveries to clients
- Self-organizing teams

Translating Agile

- Iterative Design
- Evolutionary Prototyping
- Extreme Game Development
- Scrum

Why Scrum?

- A consolidated framework
- Adopted by more and more game studios (specially those who license their technology)
- Easy to combine with different business models:
 - large in-house teams
 - outsourcing of content production

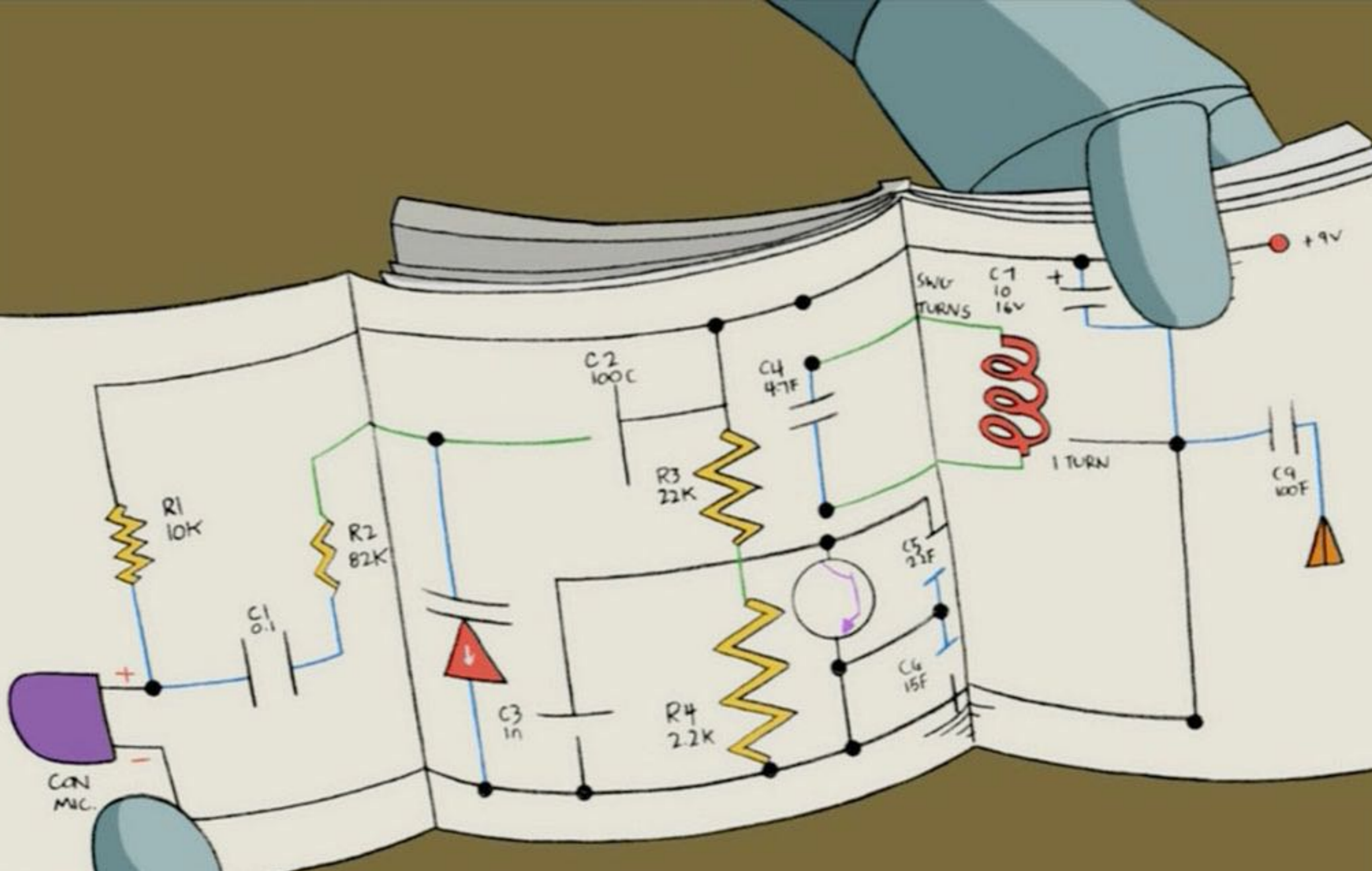




Defining Scrum

- Scrum is a framework of management and organization practices.
- An agile development method, focused on producing testable content that can be shown to a client in little time.
- A method of teamwork centered in 30 days development processes with clear goals and outcomes.
- A method for structuring work in a multidisciplinary environment.
- A collection of fancy names and jargon.

Using Scrum



Preproduction

II Production

- The product gets a **Product Owner**
- The **Product Owner** defines the **Product Backlog** and the **Sprint Backlog**
- The **Product Owner** and the Team translate the **Sprint Backlog Items** into **Stories**
- In the **Sprint Planning Meeting**, the goals of the **Sprint** are defined
- The **Sprint** starts, lead by the **Scrum Master** in the **Daily Scrum Meetings**
- The **Sprint** ends with a **Sprint Review** and a showcase of the demo

WTF!?!?!?!?!?

16 TONS

Product Owner

The member of the studio that
can prioritize the product
backlog



Product Backlog

Things the system should include



Sprint Backlog

List of tasks to perform during
each sprint



Sprint

(30 days) development cycle



Scrum master

management representative that is in each scrum team during the sprint, making sure the practices are followed and the client wishes are met



Daily Scrum meeting

short daily review meeting (15 minutes)
one speaker at a time

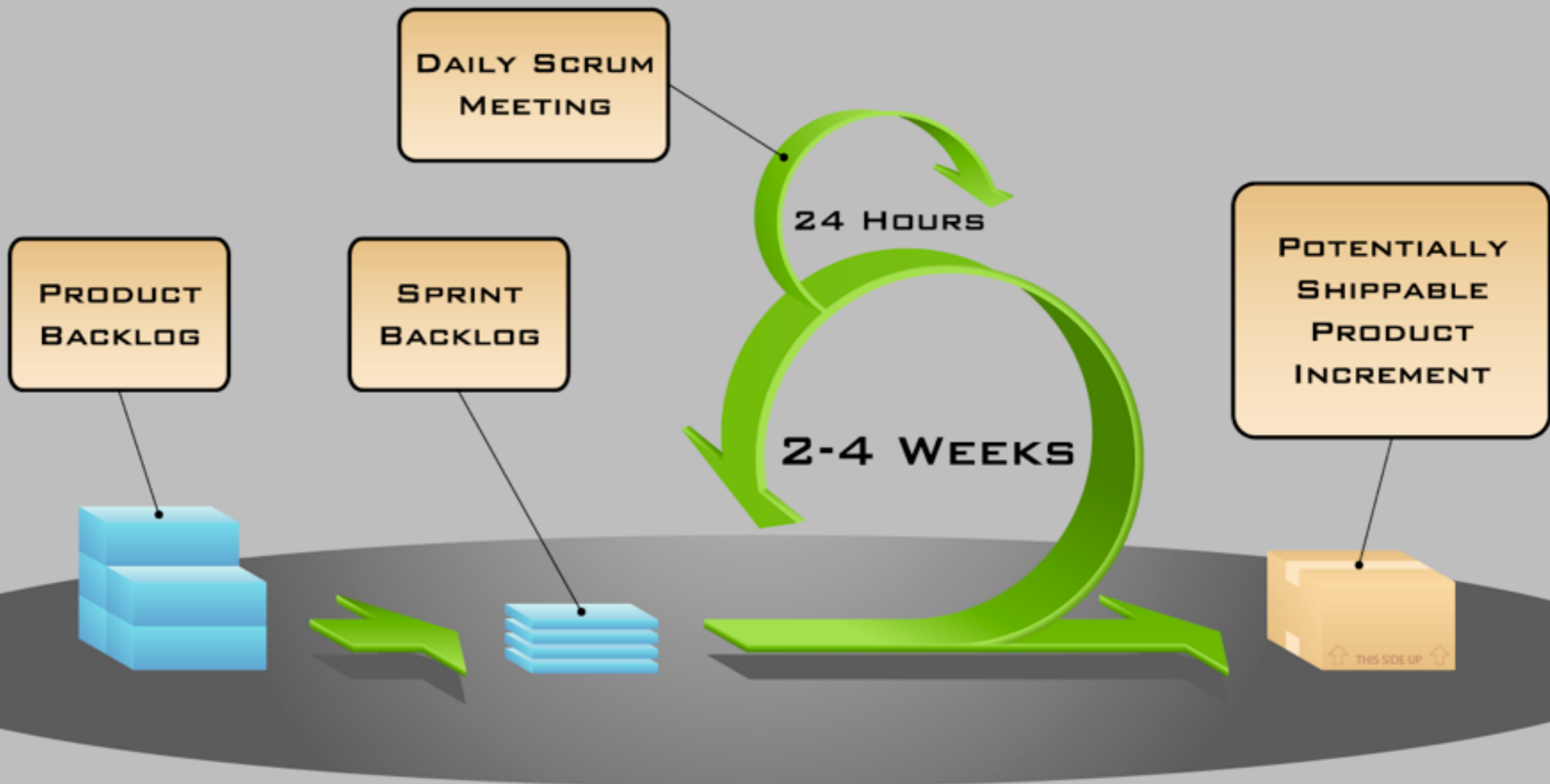


Still, WTF, but not so much



In practice

- The game is developed in short incremental stages: **vertical slices**
- Scrum teams (multidisciplinary) work together in short development periods (**Sprints**) producing a customer-focused product.
- Sprints cannot be interrupted.
- The goal is to have, by the end of the process, a working piece of software that shows functionalities.

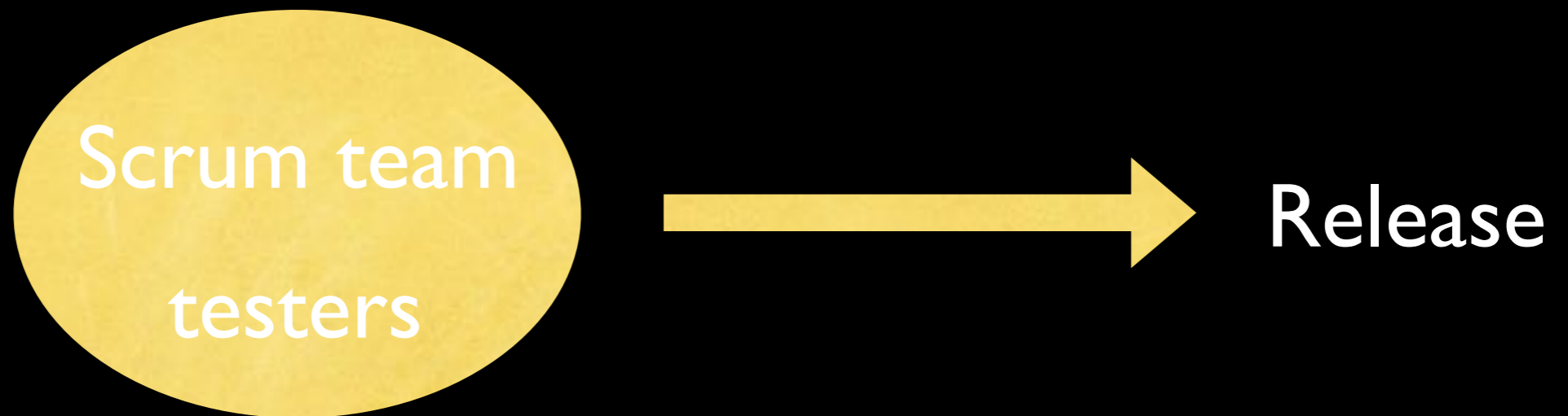


COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

The Cycle

testing?

Scrum Team ↔ Test Team → Release



How to *actually* do this?

+
i
m
P
o
r
t
a
n
c
e
-

Not
checked out

Checked out

Done

Sprint Goal

Story
- task
- task
- task

Story
- task
- task
- task



*first tasks,
then stories*

unplanned

Benefits

- Close cooperation between developer and publisher
- Lots of early testable builds: good for team morale, excellent for knowing what is wrong.
- Constant presence of players (as customers) in the development process.
- Commitment, Focus, Openness



Pitfalls

- Extremely demanding.
- Failed Sprints seriously harm development process.
- Overall picture?
- It can be easy to make poor documentation.
- ~~Funding~~.



Why do we care

- A designer knows what development methods better fit the process of creation of a particular *thing*.
- This is something you should observe in your own practice.



Summary