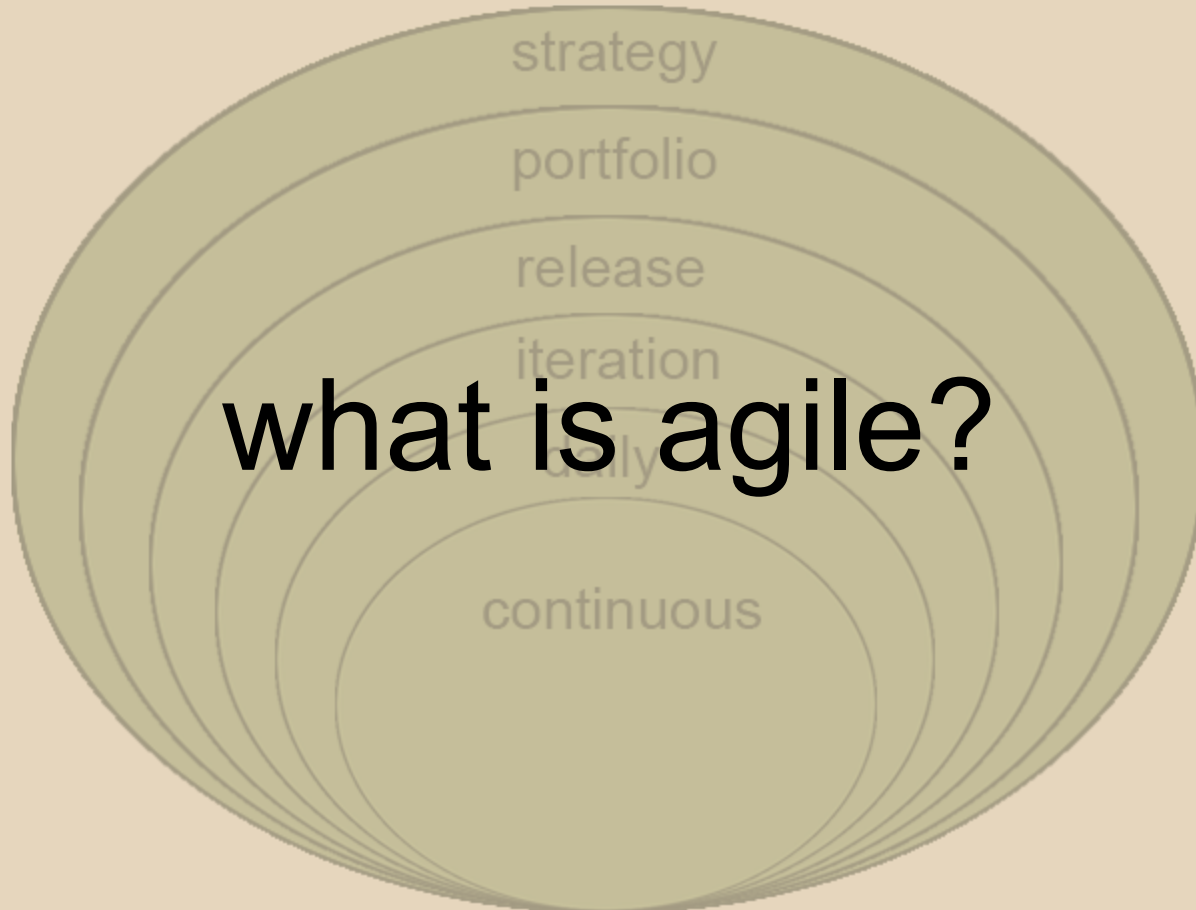




we use agile



so should you



self-organizing, cross-functional team

short development and release cycles

incremental delivery, early and often

customer drives changes

conceive

initiate

analyze

design

why agile?

construct

test

fail



avoids long-term, big-rollout predictions

delivers high-priority features early

responds to change with greater ease

eliminates waste

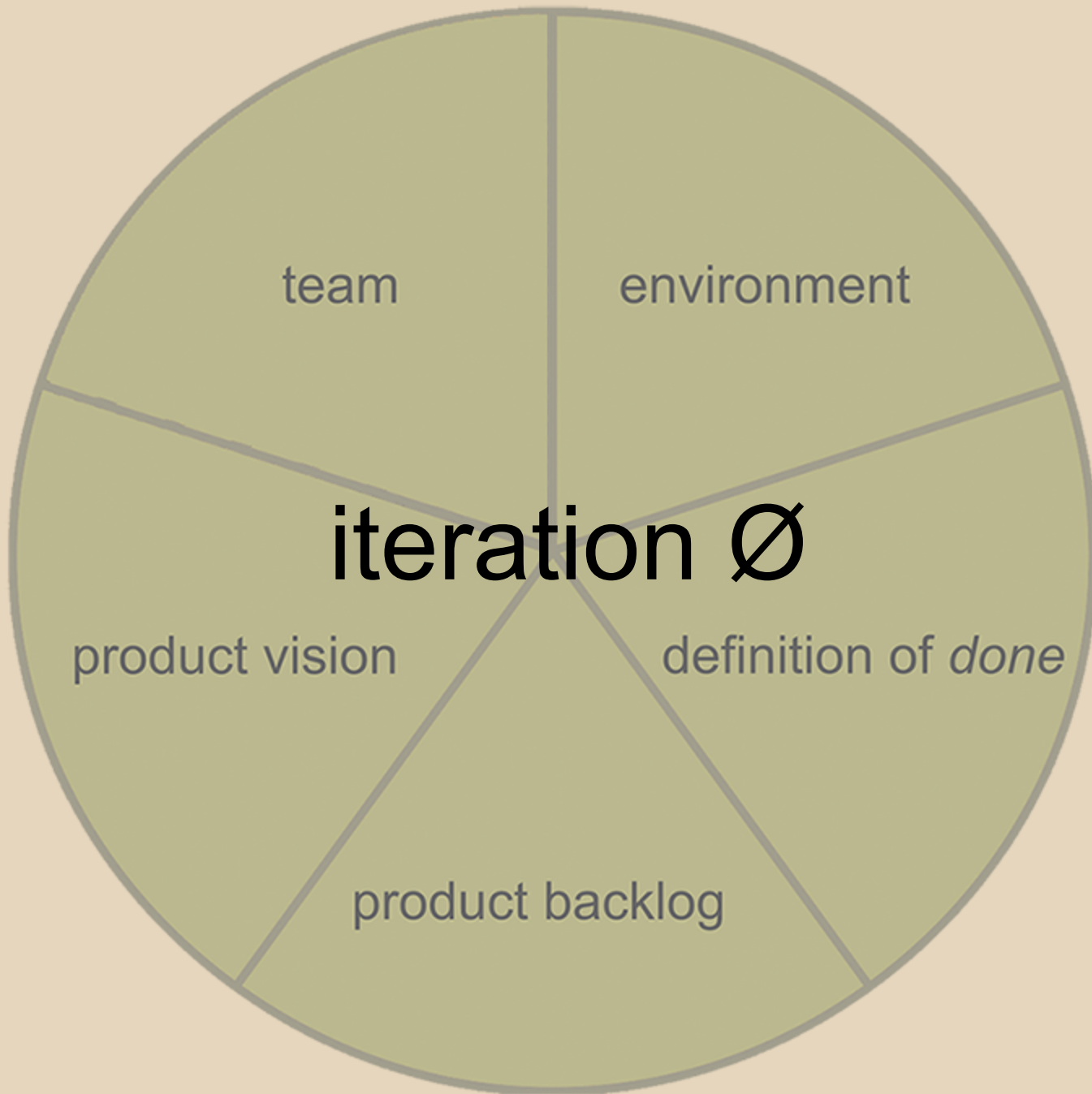
work is transparent

makes customers happier





your agile process





product vision

to provide **[product]** to **[a specific set of users]** so they can **[fulfill specific needs]**.

our product will work better than **[the alternatives]** because **[product's unique selling points]**.

if the vision isn't understood by everyone, it is not a vision.



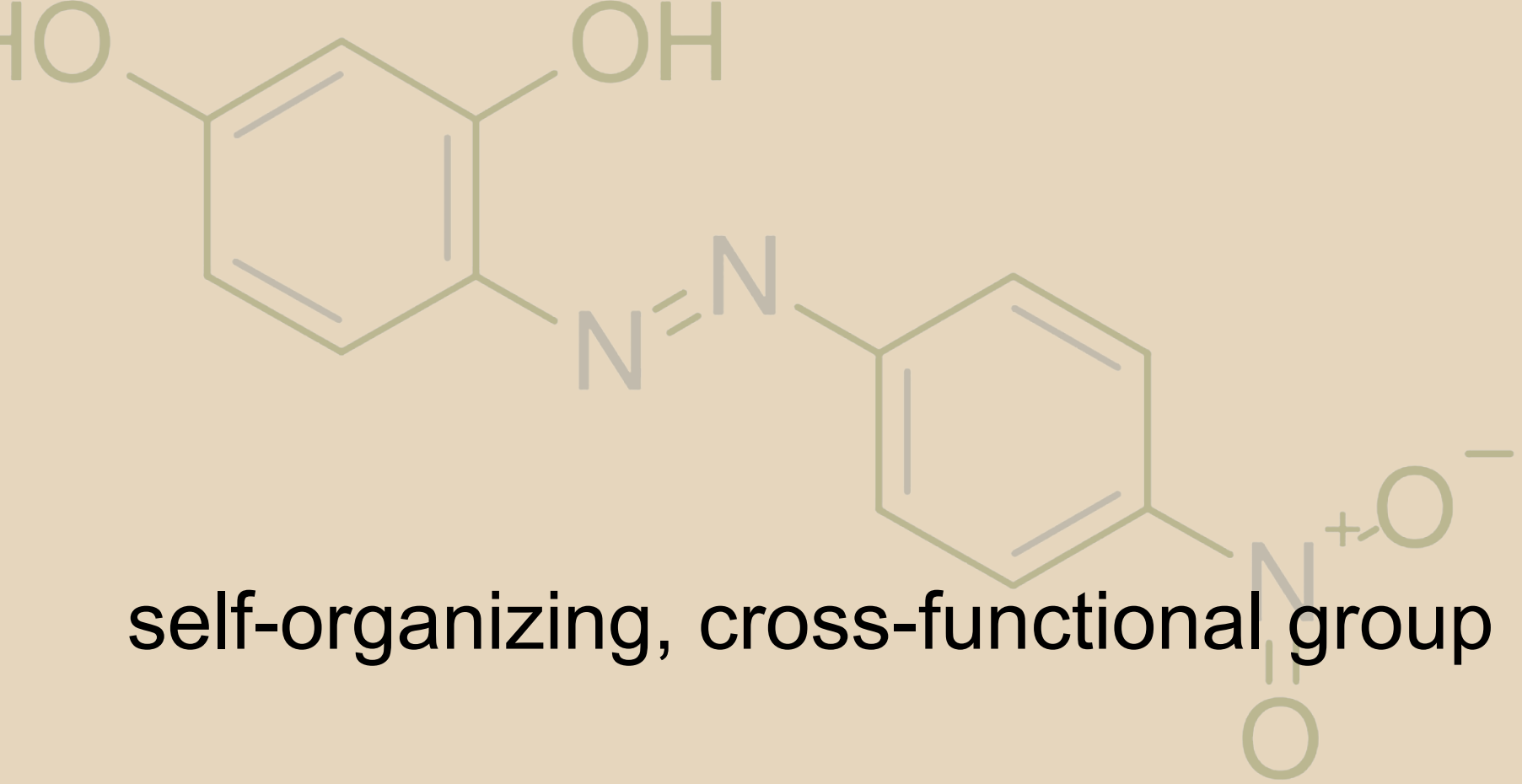




technical lead

scrum master





self-organizing, cross-functional group



workspace

tools

session logistics

ground rules for team life



team

environment

definition of *done*

product vision

product backlog

when is the { project
release
feature
story } done?

"establishing an upfront, common understanding of **done** can save teams and businesses countless hours of refactoring, process-thrash, unclear communication, and hidden work."

—mitch lacey



team

environment

product vision


definition of *done*

product backlog

incremental release of your product



features: attributes of your product



stories written for first few iterations

stories



tasks: checklist for fulfilling the story



exit criteria: definition of done



story points: 1 point "=" 4 hours of 1 person's work

user stories

as a **[type of user]**, I want **[some goal]** so that **[some reason]**.

user stories

as a **[hearing-impaired user of the system]**, I want **[to see transcripts of all the tutorial videos]** so that **[I can use their content to learn how to use the system]**.











product backlog

feature

feature

feature

feature

feature

feature

feature

feature

feature

feature

product backlog

- feature
- feature
- feature
- feature
- feature
- feature
- feature
- feature
- feature
- feature



- story
- story
- story
- story

iteration backlog

product backlog

- feature
- feature
- feature
- feature
- feature
- feature
- feature
- feature
- feature
- feature

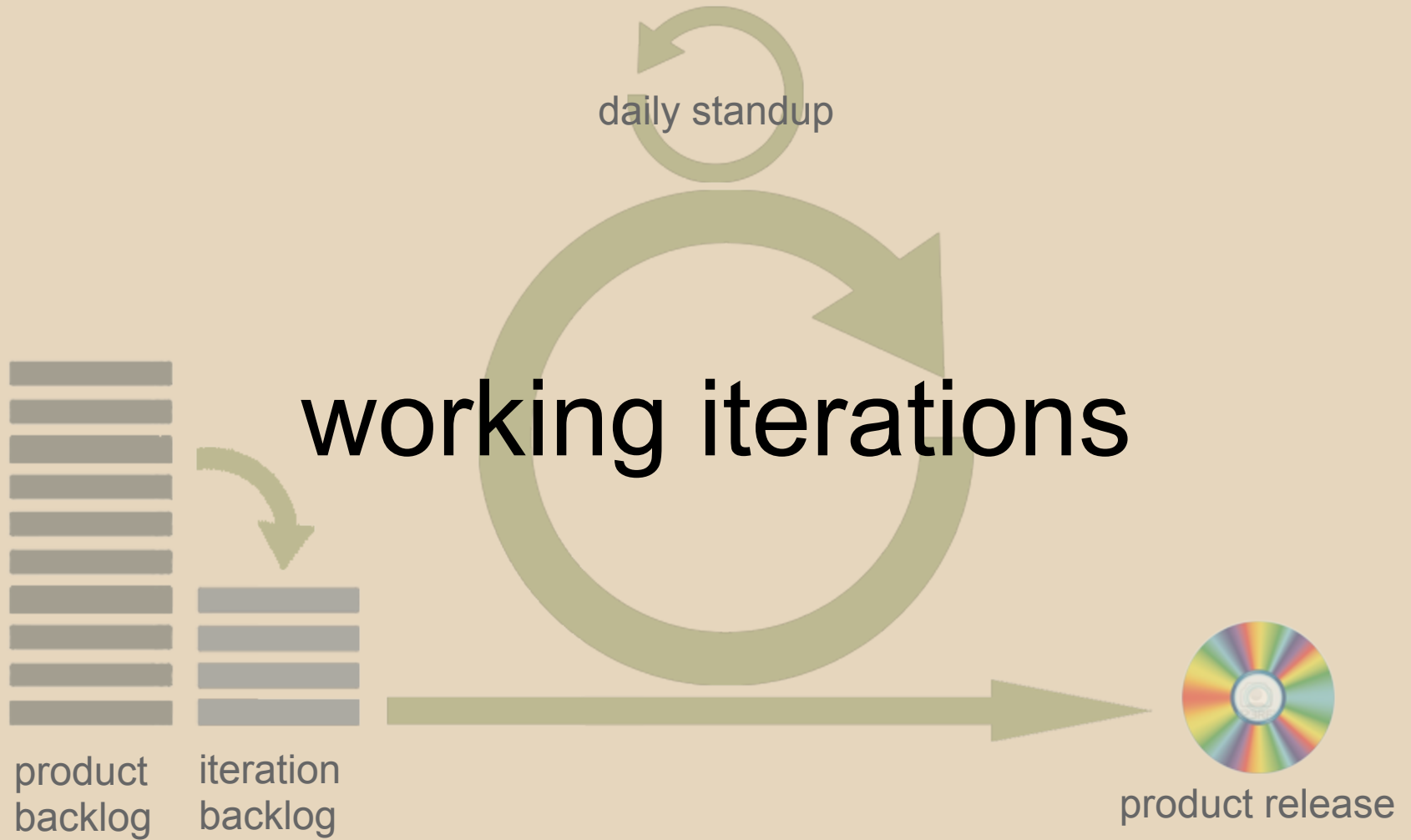


- story
- story
- story
- story

iteration backlog



what you can get done in two weeks



planning

update product backlog

build iteration backlog

analysis stories for future iterations

velocity: number of stories the team can complete in an iteration

iteration execution

self assign stories → take ownership

complete tasks

validation → exit criteria

resolve

daily standup

consistent

courteous

brief



reviews and retrospectives

review

demonstrate product

get feedback

discuss what to do next

customer may decide product is good enough for release

retrospective

not a post-mortem

identifies process improvement

start implementing improvement in next iteration

drives continual improvement.



challenges

requires practice

lack of buy-in

disputes

transparency can be threatening



no one method to rule them all

weblion's agile resources: <http://goo.gl/38qfx>

rose pruyne
weblion at penn state
rtp2@psu.edu
irc: rosepruyne