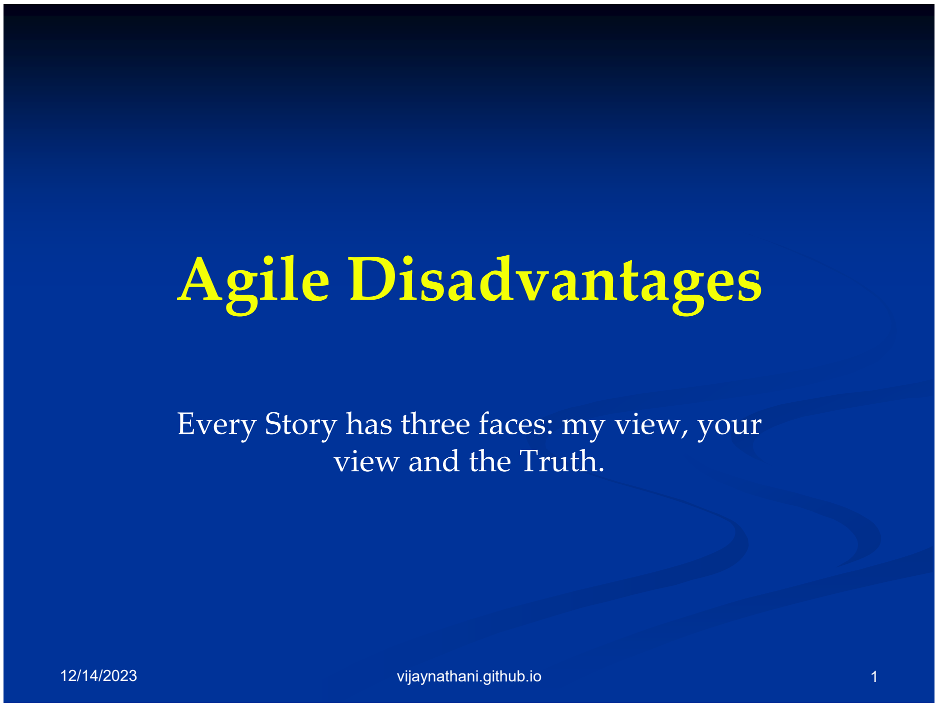
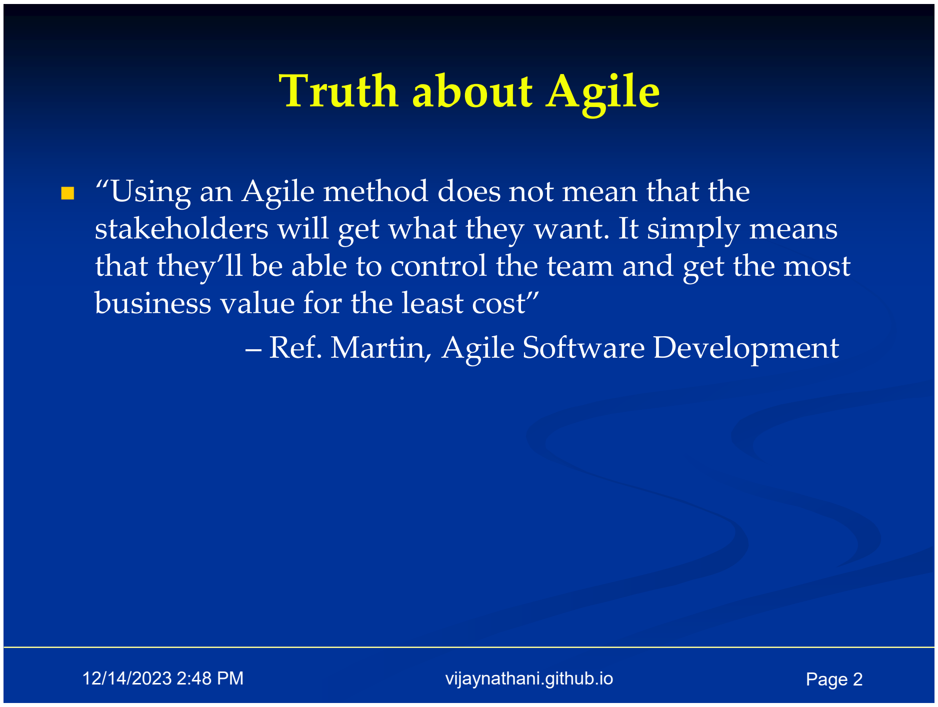


Agile Disadvantages

Every Story has three faces: my view, your view and the Truth.



Truth about Agile

- "Using an Agile method does not mean that the stakeholders will get what they want. It simply means that they'll be able to control the team and get the most business value for the least cost"

  – Ref. Martin, Agile Software Development

Lack of Skills

Fifty to seventy percent of software teams today don’t use basic, well known software practices

40% of IT shops don’t use SCM

70% of IT shops don’t even have a daily build

Developers need training. Based on my experience:

30% need basic language training.

50% need OOAD, UML, Design patterns training

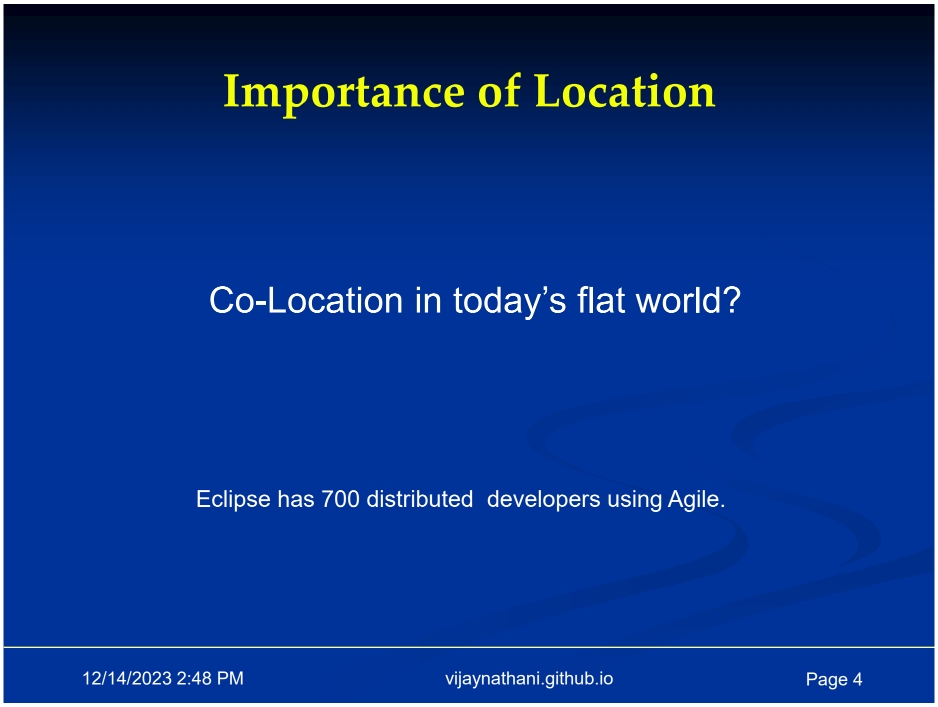
75% need automated testing training with xUnit and Fit

My preferred sequence

•OOAD, UML, Design Patterns, Automated unit testing

•Nightly build, continuous builds

•FIT. Requirements as user stories.



Importance of Location

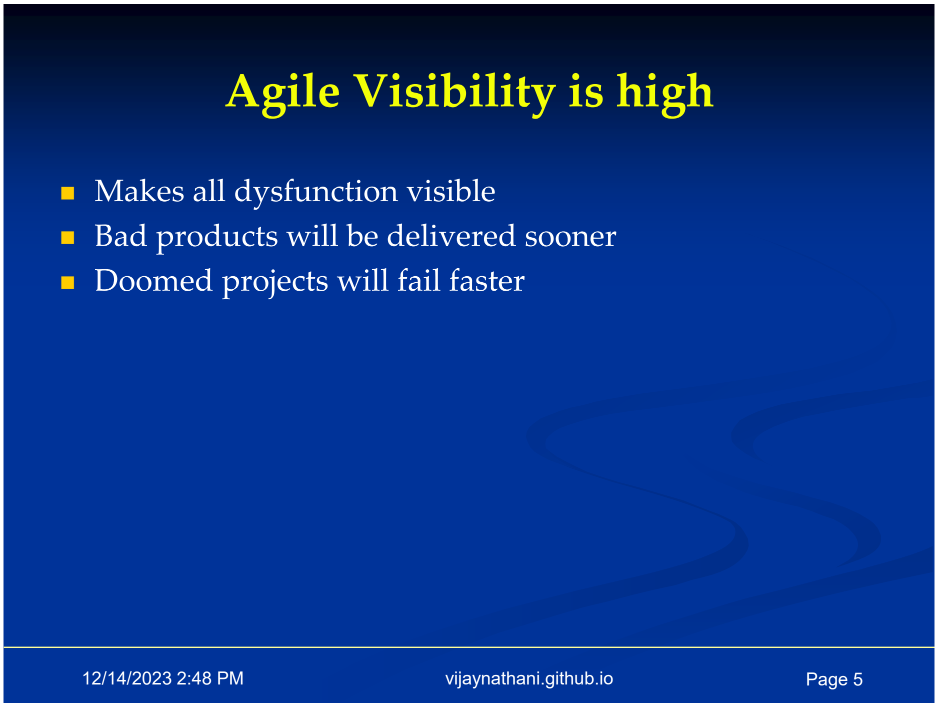
Co-Location in today’s flat world?

Eclipse has 700 distributed developers using Agile.

Business people and developers must work together daily throughout the project; the most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

> In today’s distributed, virtual world, it would be foolish to insist that all teams be co-located. However, it isn’t foolish to remind ourselves that distributed teams will never be as productive or effective as co-located ones.

Face-to-face conversations are the most effective, and distributed teams will have to work very diligently to mitigate the disadvantages of distance.

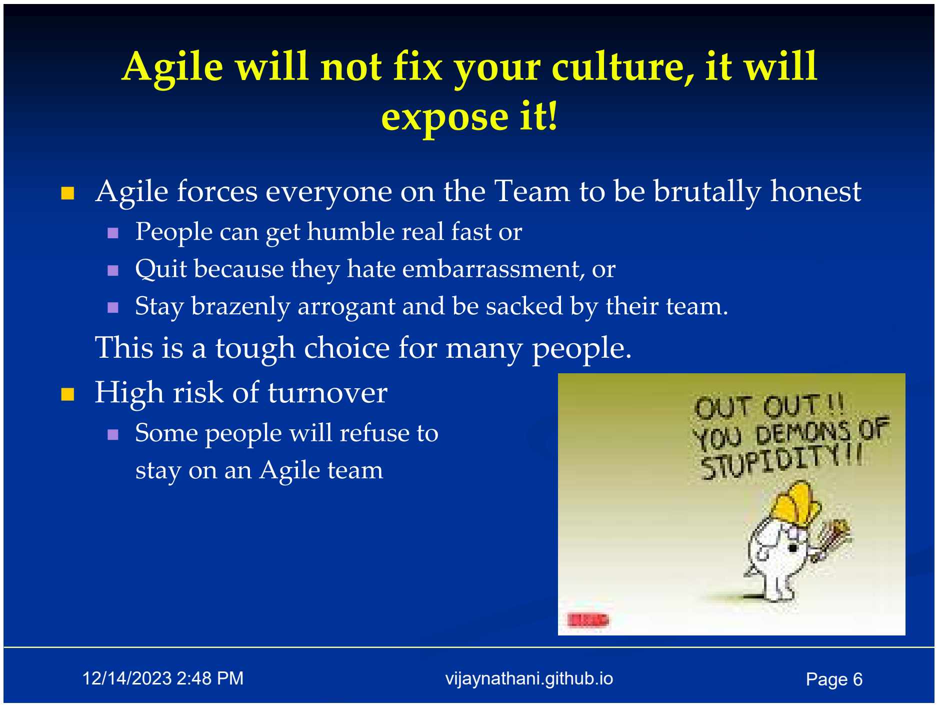


Agile Visibility is high

- Makes all dysfunction visible
- Bad products will be delivered sooner
- Doomed projects will fail faster

Agile doesn’t fix anything: the team has to do it

May feel like things are worse at the beginning



Agile will not fix your culture, it will expose it!

- Agile forces everyone on the Team to be brutally honest
  - People can get humble real fast or
  - Quit because they hate embarrassment, or
  - Stay brazenly arrogant and be sacked by their team.

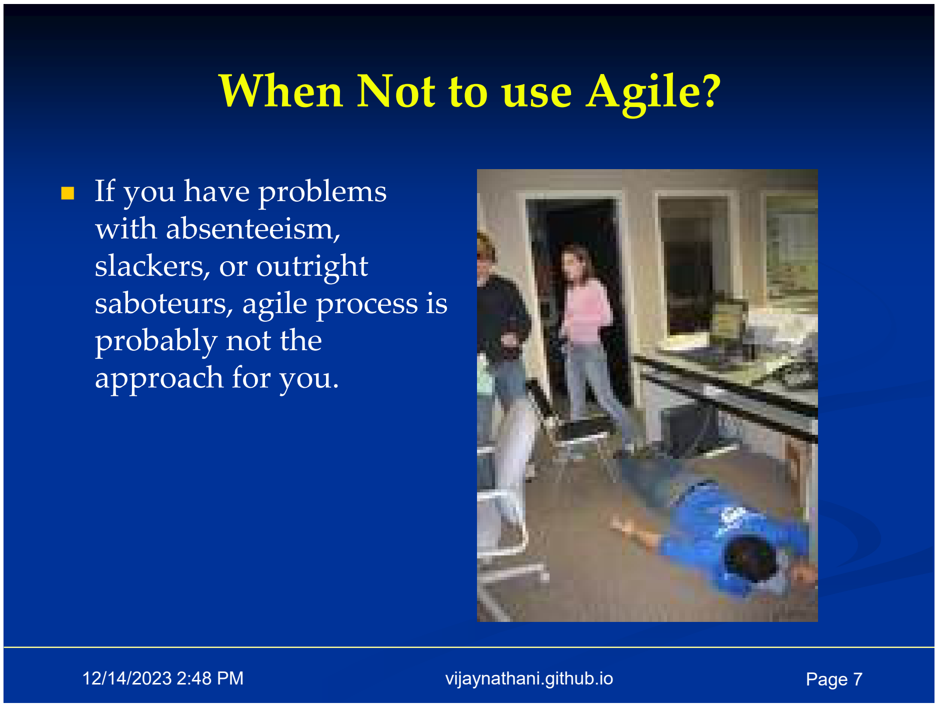
  This is a tough choice for many people.
- High risk of turnover
  - Some people will refuse to stay on an Agile team

OUT OUT!!
YOU DEMONS OF
STUPIDITY!!

In an software development company, the upper management and hard core developers usually like Agile.

The middle management usually resists agile, because of political issues. They fear loss of power. They feel that their job will be less secure, if they reveal everything to others.

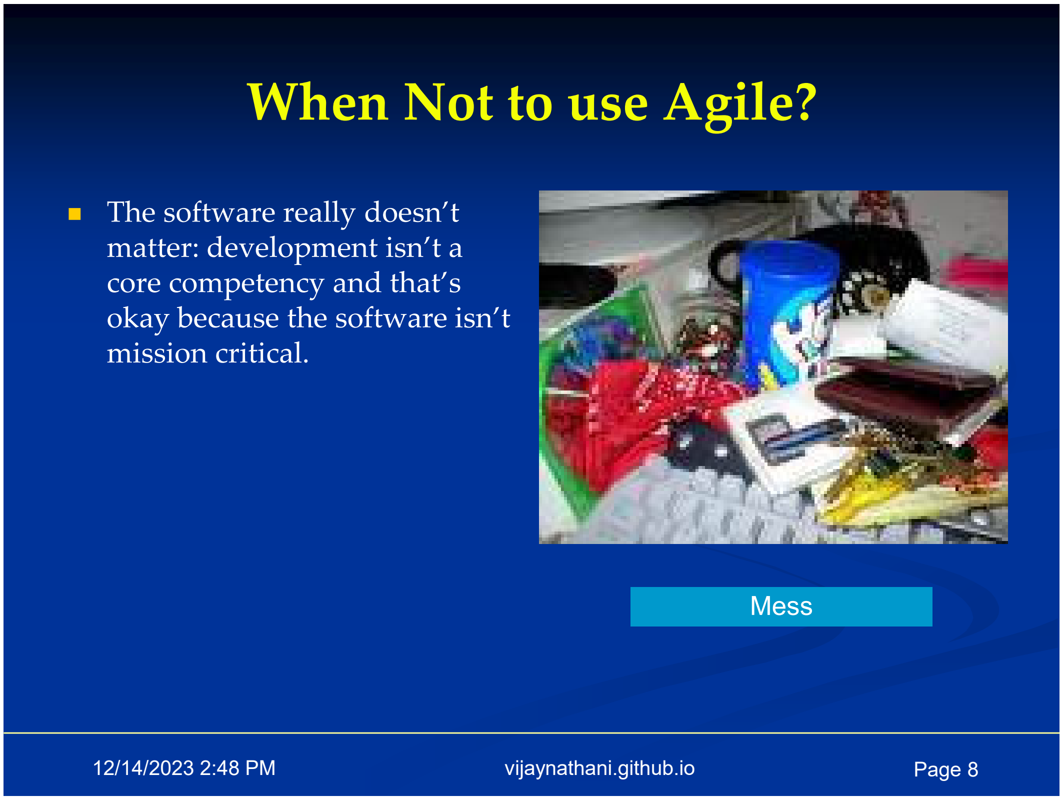
Ask the middle management, “How well things are working for you now?” Usually their current process is anyway a disaster. So why not go Agile?



When Not to use Agile?

- If you have problems with absenteeism, slackers, or outright saboteurs, agile process is probably not the approach for you.

You’ll need something more heavy-handed, slower, and less productive e.g. Command & Control Management.



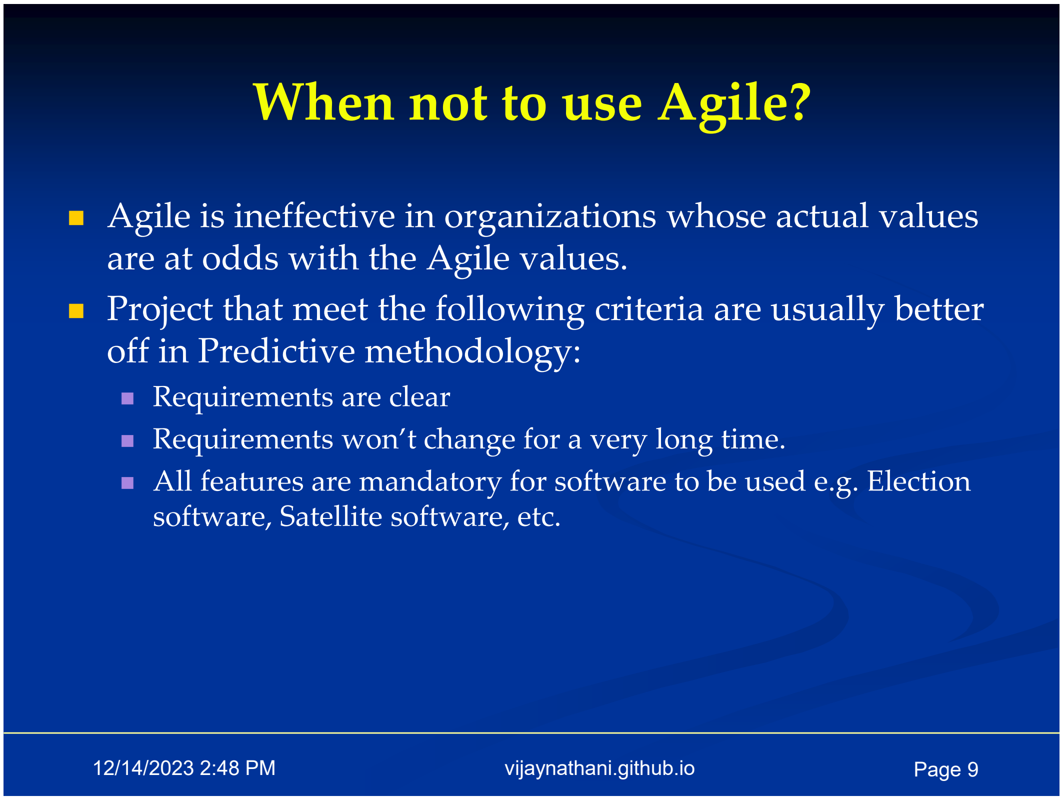
When Not to use Agile?

- The software really doesn’t matter: development isn’t a core competency and that’s okay because the software isn’t mission critical.

Mess

Your engineering practices embrace heavy, up-front design, the construction of baroque frameworks, and throw-it-over-the-wall attitudes towards QA.

Nobody can agree on ‘done-ness’



When not to use Agile?

- Agile is ineffective in organizations whose actual values are at odds with the Agile values.
- Project that meet the following criteria are usually better off in Predictive methodology:
  - Requirements are clear
  - Requirements won’t change for a very long time.
  - All features are mandatory for software to be used e.g. Election software, Satellite software, etc.

I say "actual values" because many organizations have professed values differing from or contradicting the values revealed by their actions.

> E.g. If an organization's actual values are secrecy, isolation, complexity, timidity, and disrespect; suddenly expressing the opposite values through Agile will cause trouble rather than create improvement.
>
> E.g. Your management practices embrace ‘do it now and forget what I told you to do yesterday’.
>
> E.g. Your company prides itself on being ‘flexible’ and ‘adaptive’: Actually, these are code words for lack of discipline.

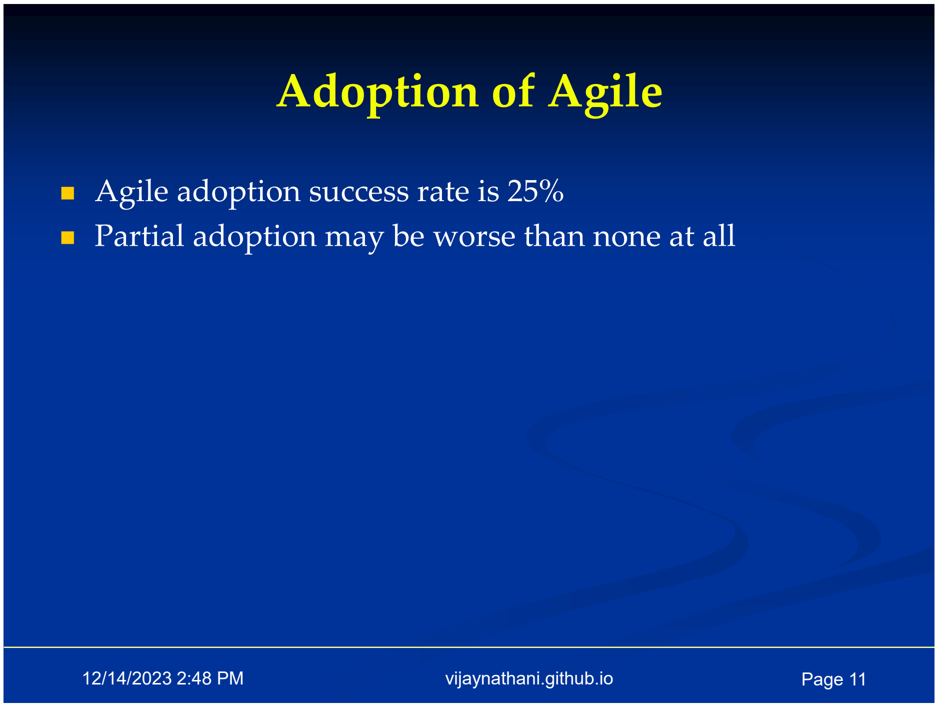
Agile can Hurt

- Agile exposes flaws in the Organization.

Here's a sad but repeated story: a development team begins applying Agile, dramatically improves quality and productivity, but then is disbanded, its leaders fired and the rest of the team scattered. Why does this happen?

Answer: The team's improved performance shifted the constraint elsewhere in the organization. The new constraint (e.g. marketing, who can't decide what they want fast enough) doesn't like the spotlight. Nobody actually cares about organizational throughput. The "source" of the turmoil, Agile, is blamed and eliminated.

==



Adoption of Agile

- Agile adoption success rate is 25%
- Partial adoption may be worse than none at all

Of every 3 companies that attempt to use Scrum, 1 actually uses it. The other two may use certain practices but never adopt in its fullness.

Agile fails not because it has flaws but because of people fear of transparency. It takes courage to be truthful.

If adoption fails, time will have been wasted, and some people may leave

When people say partial adoption it usually means that

•if time permits, I will do the testing.

•Sometimes the manager assigns work to the team. Sometimes the team is self-managing.

Both are disastrous for the project