Code quality is the most important thing

Team members have
a natural pecking order

A manager's job is creating a supportive environment

A manager's job is maximizing the work their people do

Programmers write code;
testers break code

The Product Owner is too important to get involved in the details

The whole team succeeds together or fails together

Every bad event is an opportunity to learn

I know what the users *really* want

It's not my job to make decisions.

Programmers should do exactly what the Product Owner asks, and nothing else

We should not build
unethical systems,
even when ordered to do so

If I lose this job, I might not be able to get another

I can always get another job somewhere else

If I don't say "YES," people will say I'm not a "team player"

We can't take action
until we all agree

What happens at work doesn't really matter; it's just a means to a paycheck

An Agile team
doesn't need managers

Things will get done when they get done

We are stronger, together, when we have different viewpoints

We go faster when we all see things the same way

Things will go better
if people just
leave me alone

What has worked in the past
will continue to work

Managing work in an electronic tool is the most efficient

Efficiency is a top priority

If we don't get it "right,"
it doesn't matter how
efficiently we work

If we don't get the architecture right at first, we'll never be able to recover

Software architecture, unlike that for buildings, is flexible and can be easily changed

User experience design needs to be substantially complete before we can start development

We should test our user experience with real users as we develop

Development needs to be substantially complete before we can test it

Automation isn't testing

We can test our intentions
before we write a line of code

Programmers shouldn't have to test; that's what testers are for

Programmers can't be trusted
to test,
especially their own code

We shouldn't start designing
the software until the
business analysts collect
all the requirements

We'll never know all the requirements without building something people can experience

Programmers and testers are
like cats and dogs,
or oil and water

We're not here to make friends; we're here to get stuff done

Good work is the result of having the best people

Bad work is the result of having bad people

When something goes wrong,
it's important to
hold accountable
the person responsible for it

What we need is
a programmer whose skill
level in the implementation
language is at least
a 9 out of 10

What we need is a programmer who learns quickly

What we need is a programmer who works very carefully

What we need is a
programmer who
works well with others

When building a team, it's important to fill the key roles

When building a team, it's important to get the highest performing individuals

When building a team, it's
important to create
a sense of interdependence

When building a team, it's important that the members all like each other

When building a team, it's important that
there is no conflict

When building a team, it's important that they learn how to navigate conflict within the team

When building a team, it's important that they stand united against outsiders