

Build Process Patterns

Automate only where necessary

- build for value

Process defines itself

- reuse proven tasks

Task defines itself

```

\define thread
of control

```

```
define
required
state
```

```
define
task
name
```

```
define
exceptions
```

define
collaboration

Task Collaboration Patterns

Contact us

data for
response

access
argument
for task

Complete the form

Record information

pass
argument
to task

- Define process flows

```
configure
handling task
```

Escalation

```
handle
-exception
```

Do your work

Hand-off

pass task to
different team

handle
overload

Pick up task again

notifies of
completion

Task complete

indicates
complete

share state
between tasks

configure
handling task

Do your work

- reduce threading overheads

Hand-off

Task Execution Patterns

Manage Team Size

manage number
of threads and
resources

Responsible Team

- associates
- responsibility

Team

detect
overload

Team too busy

Do it yourself

- reduce
—threading
overheads

- executes categories of tasks

Categorise Task