Team Level Up workshop – the cards

Version 4 – updated: 5/22/2025

Note: See Team Level Up folder
File: Master – Team Level Up – Simpler Version

The Wisdom of Teams
- A Real Team
- Small in Number, <10
- Complementary Skills
- Committed to Common Purpose
- Performance Goals
- Agreed Common Approach
- Mutually Accountable

IDEO – HOT Team
- Totally dedicated to end result
- Slightly ridiculous deadline
- Team irreverent and non-hierarchical
- Well-rounded and respectful
- Eclectic space
- Flexibility
- Group Work
- Brainstorming
- Empowered to get whatever needed

The New New Product Development Game
- Built-in instability
- Self-organizing project teams
  - Autonomy
  - Self-transcendence
  - Cross-fertilization
- Over-lapping development phases
  - Cf Waterfall
  - Sashimi approach
- “Multi-learning”
  - Multi level
  - Multi functional learning
- Subtle control
  - Control through peer pressure

    - Self control
    - Control by love
  - Transfer of Learning
    - Drive to learn
    - Drive to share learning
    - Create std practice

- Levels of Work
  - Epic
  - Feature
  - Sprint-Sized User Story
  - Task
- Why break down?
  - Need bite-sized stories
  - Faster Feedback
  - More transparency re progress
  - Know when to ask for help
  - To sopprt a game, with Wins
  - To enable faster learning
  - Better plan, better date guess
- Goals in breaking up stories
  - Achieve 80-20
  - Less work, more BV
  - Chunks that enable Team productivity in building
- Epic, Feature, User Story -> have BV too
- Tasks are only work

- Mapping (matrix) of Epics, Features, Stories into:
  - Sprints & Releases
  - Multiple planning purposes
  - "When do I get to use my Feature?"
  - Find best way to build & deliver BV
  - Aka Sprint Mapping

END COLUMN 1

General

We want the following; ideally these were clear at the start.

One team –> One vision (one product goal)
Don't get distracted
We have a Vision & Mission that is important
Well and commonly understood
Team members are Team Players
Team inspirable
Team inspired
Team is Full Time
Team thinks they might win
Have sufficient skill sets
Impediments are ok
Support is ok
We are set up for success
Team members want to win, together
Game with Rules
Complementary skills
No SPOF (single point of failure)
Less Silo'd with time
Help each other
Ask for Help more, sooner
Learn for next time
Plenty of Impediments, but no showstoppers

List our impediments (subset could double Velocity)

We want more fun
The best 6 months of your career

All Scrum Rules in SG (vs)
Most Scrum Rules

A bare framework
Framework incomplete, we must add
BUT: process still KISS
Minimum Viable Bureaucracy
So, in some sense, we are doing more than Scrum

Empirical Process
Transparency, Inspect and Adapt

- We are always going to be getting better
- We are willing to try small experiments to get better

Scrum Values:

- Commitment
- Focus
- Openness
- Respect
- Courage

Roles, Meetings, Artifacts

Roles

- PO
- SM
- Developers
- Chickens
  - Business Stakeholders
  - Minions
  - Others
- Team size = 7
- Scrum team members 100%
- Each member has 1 role
- Each person set up to perform role decently

Events

- Sprint
- Sprint Planning
- Team commits to reasonable amt of work
- Daily Scrum
- Sprint Review
- Retrospective

Artifacts

- PB/PBL
- PB Items (PBIs)
- Product Goal, Vision
- SBL
- Sprint Goal
- Scrum Board
- Sprint Burn Down
- Release Burn Down
- Working Product / Increment

- DOD
- DOD stronger soon (eg, zero bugs)
- Impediment List

Other Things

- User Stories
- Story points
- Track Velocity each Sprint
- Expect Velocity to be going up
- Expect 25% increase in 3 months
- DOR
- DOR process

PB Refinement

- Short-term Mtg
- Long-term Mtg
- Expectation that Plan will change

Team / team member WIP limits during Sprint

END COLUMN 2

Team Norms

- Time for Daily Scrum (eg, 9:15am)
- Start – End Hrs
- Core Hours
- How Communicate
  - General Methods (in-person, in writing, email, phone, text, Slack, etc.)
  - Approaches per person (conversation, picture, lists, conceptually, with examples, Specifics, etc.)
- Vacation Days

Decision-making

- We have agreed on one method
  - Time-box: 30 mins
  - ID best decision-maker
  - All discuss (within timebox)
  - Decider decides quickly
  - Different if costly decision
- We have alternate methods

PO gathers input re PBL order
PO decides order
How should other decisions be made?
All talk: one decider, quickly
Other method(s) of deciding

***

Other
Min Num of stories per sprint
All roughly equal in size

Other Decisions
How to do code review
When to fix
Which Tech stack
TDD?
Pair Programming?
Other parts of XP?

Ideas
All of one, one for all
Accountable to each other

Sprint Duration
2 weeks
Consistent cadence

Sprint Commitment
At least 8 PBIs per sprint
Size of stories in Sprint (avg size & range)
Def of Reliable
Each person willing to support any stuck story or task

Sprint carryover-handling
Not counting stories added if space at end of Sprint

Business Stakeholders – 20%?

Manage Chickens
Who can Chickens be?
What help does Team need
Identify specific chickens
Work with them

Manage them
Escalate when needed

Metrics & Similar
Velocity
Carryover - # of stories
How handle?
Happiness
BVPs
Bugs escaping Sprint
Quality – how?

We expect Velocity to increase 100% by ?

Code reviews
Fix all Code review issues in same Sprint (DOD)
Velocity graph

Using Impediment List
All types of Impediments
Add items to Imped List continuously
Team works on Impeds via Sprint Bklog (some)
Expect Managers to help (some)
Order: Mainly ROI + Depend + other
Team will prioritize impediments
Track fixed impediments

Team skills gaps
Managers will ID and propose fill
Team will ID and propose fill
When? (Start, S3, every 4 sprints)

Other
Coding Standards
Team members help each other
Team members ask for help sooner
Team members accept help

We define product success at release points
We measure success
Work on top problem now

Performance approach

- 50% team success
- 50% individual contribution

Other

- Transparency, Openness, Honesty
- Discussion of issues
- Fail and Learn
- Experiment
- Some level of failure is good
- Be honest about it

- Willing to do whatever it takes
- Make clear in DS if stuck
- Person can ask for more help, up to everyone

- Share ad collaborate
- Pairing
- "Missouri": 15 mins of sharing / 1/week
- Tech PIT – tech issue, 1 hr max

- Use timeboxes for work
- Be reasonable
- 4+ hrs / day (focus)
- Max 40 hrs week (clock)
- 2 hr tasks

Minimize distractions

- For everyone
- Key SM role
- Pomodoro technique
- List common distractions
- Other?

END COLUMN 3

Other

- Stories done in ~2 days
- QAs/coders ratio
- Tests automated 80%
- QA tests cover much of UAT
- Test First
- Mini Sprint each day

- Good DOD?
- All bugs fixed?

- PO decides order
- PO decides MVP
- PO ans Q's in Sprint quickly
- PO respo for DOR process
- All Q's answered before Sprint start
- PO shoots for 80-20
- PO should incr. BV per Release 50% 1st year

- Team knows 20 Ways to break up stories

Use of Scrum Board

- Min WIP
- Attack any stuck item
- Update "hourly"

Other visual management techniques

How practiced is everyone? [??]

Values. Principles, Ideas

- Scrum Values
  - Commitment
  - Focus
  - Courage
  - Respect
  - Openness

- Agile Manifesto
  - Individuals & Interactions > Processes & Tools
  - Working SW > Comprehensive Documentation
  - Customer collaboration > Contract Negotiation
  - Responding to change > Following a plan

- Agile Principles
  - 1 – Satisfy customer
  - 2 – Welcome Changing Reqs
  - 3 – Deliver Freq
  - 4 – Work together daily

5 – Motivated Individuals, trust
6 – Face-to-face

7 – Working [prod] prim meas of progress
8 – Sustainable development
9 – Tech Excellence & Good design
10 – Simplicity, work not done
11 – Emerge fr self-organizing teams
12 – Reflect, tune, adjust

Ideas

The bad news doesn't get better with age
If you wait for perfection, you might wait too long
Fail fast, learn faster
When you come to a fork in the road, take it
Shut up and drive (Rihanna)
The Krispy Kreme theory
People are remarkably good at doing what they want to do (JL)

All see the same elephant
Be disciplined.

More fun / happiness
Accountable to each other
Have fun regularly

Team eval at start: can we win?

Events

Sprint Planning

Create tasks in SPM (not before)
Stories & Tasks
DOR, right?
"Speak Now"
SP on all Sprint Stories

Daily Scrum

Daily
Max 15 mins
Biggest impediment

    - Say what helps Team
    - Say what helps the Team
    - Everyone listens
    - Purpose: Be more successful
    - Emergent leadership

  - Sprint (itself)
    - Working product by end of Sprint
    - Working product “each day” in Sprint

  - Sprint Review
    - Review (of situation)
    - Demo of Working Product

    - Good “customer” feedback (review)
    - Full feedback
    - Everyone can give feedback
    - Honest negative feedback
    - PO resolves disagreements immediately

  - Retrospective
    - Positives
    - Negatives
    - SM Report
    - ID top (4) impediments
    - Use A3 approach

- Artifacts
  - Product Backlog
    - Stories for 3+ Sprints are sprint-sized
    - PB goes out 1 year (ish)
    - Later items are larger
    - Goal: 80-20
    - Use INVEST (Indep, Negotiable, Valuable, Estimable, Sized Approp, Testable)
    - Transparent

  - Top 20 Impediments
    - Public
    - Helps
    - Prioritized
    - Track when created
    - Work 1 at time
    - Small (fixable in 1 Sprint)

End of column 4.

Clearer Req’s

- Verbal, written, pictures, etc.
- DOR
- DOR Process
- Devs expected to reject stories
- Minions help
- Document only what’s needed
- New process helps
- What works for this Team

Other

- PO pivots when needed
- PO also maintains course when needed
- Team keeps an open mind re direction
- Stronger, more automated testing
- UT, FT, INT/REG Test
- Other Test Types?
- Fix ALL identified bugs (almost)
- Better continuous integration

Product Backlog

- First things first
- Prioritize mainly by ROI
- Include everything
- Prioritize everything

- Single piece continuous flow

Planning

- Trust and professionalism in planning
- Adaptive planning
- Plan’s always getting better – less inaccurate
- Plan’s sometimes getting better for customer.
- Pos & Neg change
- Customer collaboration > contract negotiation

- Doing “everything at once”
- Actively manage PB
- Build knowledge
- Build knowledge JIT
- Build our customer understanding

- Increase self-organization
- More collaboration
- Better collaboration
- Swarming
- Other adaptations intra-sprint

- Inc. rate of innovation & creativity
- Build in more innov & creativity
- Define specific ways I&C will happen
- Discuss diff levels (eg, code, module, systems design, implem., bus sol, etc)

- Rate current recognition level
- ID more ways to recognize
- Give more recognition

- Rate morale of Team
- Include inspiration / motivation
- Include pos & neg
- ID how to improve morale
- Experiment

- ID team member strengths & weaknesses
- ID ways to improve
- ID ways to help them
- If a team member can't cut it, help that person transition out

Top 20 Impediments List

- Share & collaborate across teams

Build collaboration

- With Business side
- How?
- Why?
- Better requirements (details)
- Less wasted energy
- Faster delivery
- More BV
- Higher morale

Build Team

- More "buzz" in team
- We will build it - How?
- More rhythm, harmony

End of column 5

- Agile Release Planning
  - Planning over Plans
  - Time is Important
  - Smarter, Not Harder
  - More accurate plans (as we progress)
  - Manage pressure

- Key Activities
  - Bring Scrum Team & BSHs
  - One day for 6 mos.
  - Vision
  - Product Backlog
    - Roles (5-7)
    - User Story Workshop (50-60 for six mos.)
  - Business Value
    - Drivers (3-5)
    - Priority Poker (BVPs)
  - Effort
    - DOD review
    - Planning Poker (SPs)
  - ROI = BVP / SP
  - Other factors
    - Risks, Depend, Learning, MMFS/MVP, other
  - Identify Expected Velocity
  - Lay out stories into Sprints
  - Identify # of Sprints for 1st Release
  - Add contingency
  - Add Landing Strip
  - Calc avg cost per Sprint
  - Calc budget for 1st release

  - Key Goal: prioritize our stupidity
  - Learn! (to then improve plan)

- Revising the Plan
  - PB Refinement
  - ST Meeting
    - Vote re details for each story
    - Assure DOR good before SPM
  - LT Meeting
    - Many activities to revise and refine the LT plan

Anything in ARP can change, eg, Vision Stmt
Mainly: new stories, broken up stories, new BVPs, new SPs
Then: consequences
Try to make the plan better (more accurate)
Try to make the results better for customer

Misc

Stop tracking worked hours to assigned stories
2 hour tasks
Re-estimate remaining hrs on tasks
In DS, must say what you did and what you will do
SP on all stories
See Velocity of Team
"Velocity" (mainly) avg over last 3 sprints
Coders and testers playing same game
Team makes sure work is balanced across individuals

We are all in it together
Team compares SP to task estimates in SPM

Happiness metric
See Jeff Sutherland's blog – scruminc.com