National Snow and Ice Data Center

Supporting Cryospheric Research Since 1976

The Human Side of Agile in Earth Sciences Application Development

Including, but not limited to:

Julia Collins, Danielle Harper, Margaret McNulty, Jess Lacy, Luis Lopez, Scott Lewis, Miao Liu, Joe Oldenburg, Stuart Reed, Donna Scott, Ian Truslove, Hannah Wilcox

National Snow and Ice Data Center
Cooperative Institute for Research in Environmental Sciences, University of Colorado

UCAR Software Engineering Assembly
21 February 2012

About the National Snow and Ice Data Center

- NSIDC manages and distributes scientific data, creates tools for data access, supports data users, and educates the public about the cryosphere.
- Research faculty at NSIDC address all aspects of the cryosphere: snow, ice, glaciers, frozen ground, and related climate interactions.

If it involves frozen water, we're interested!

About Us

Developers of:

- Data management tools (often involving browser-based GUIs).
- Web portals for data access.
- Scientific data products.

About Us

Technologies:

- Database:
  - PostGRES/PostGIS (moving off of Sybase)
  - Solr
- Java, Hibernate, Struts
- Ruby on Rails
- JavaScript
- Perl
- Python
- IDL

History: Project Management

- Projects conceived and implemented independently from each other.
- Developer(s) assigned to a particular scientist or product team.

History: Project Management

Impacts:

- Difficult to achieve cohesiveness in software development practices.
- Did not foster code re-use.
- Dependencies due to limited number of people with knowledge of a project.

Ultimately:

Frustrated software developers and a few frustrated project managers.

History: We want to be "agile"

- Fresh ideas brought in with new employee(s).
- Inspired by Alaska Satellite Facility (ASF) successes.
- Had a new project coming online that we decided to use as a pilot study for the process.

The result...

- Were we really "agile"? Were we doing "scrum"?
- A few people (both managers and developers) studied some more, decided we weren't, and drove the next steps.

Donna + Brendan + book = Inspiration

Next step: Consult with someone that knows what they're doing.

(We used up the training budget for group training with external Scrum Coach.)

Transition

- Identified projects willing to participate in the process.
- Project products ranged from new browser-based applications to scientific data products.
- Decided to start with two teams and split the projects between the two.
- Identified team members.

Continuing Education

- Second training session for those unavailable during the first series.
- Separate training session for Product Owners.
- Subsequent new hires learning as they go.

Definitions: Scrum

Scrum (n): A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

The Scrum Guide, Jeff Sutherland and Ken Schwaber, 2011.
http://www.scrum.org/storage/scrumguides/Scrum_Guide.pdf

Definitions: Scrum Roles

- Product Owner (PO): The person responsible for the project's business value.
- Scrum Master: Maximizes the value of interactions between all parties.

Definitions

- Sprint (Iteration): a constant length time interval for work, typically 1-4 weeks. (2 weeks @ NSIDC)
- Story: a valuable slice of new functionality.
- Story point: unit in the scale of effort required to complete a story.
- Velocity: average rate of progress through work (story points per sprint).
- Backlog: a prioritized list of stories that represent the work to be completed.

Practices

Scrum teams

Starting point: "Book scrum"

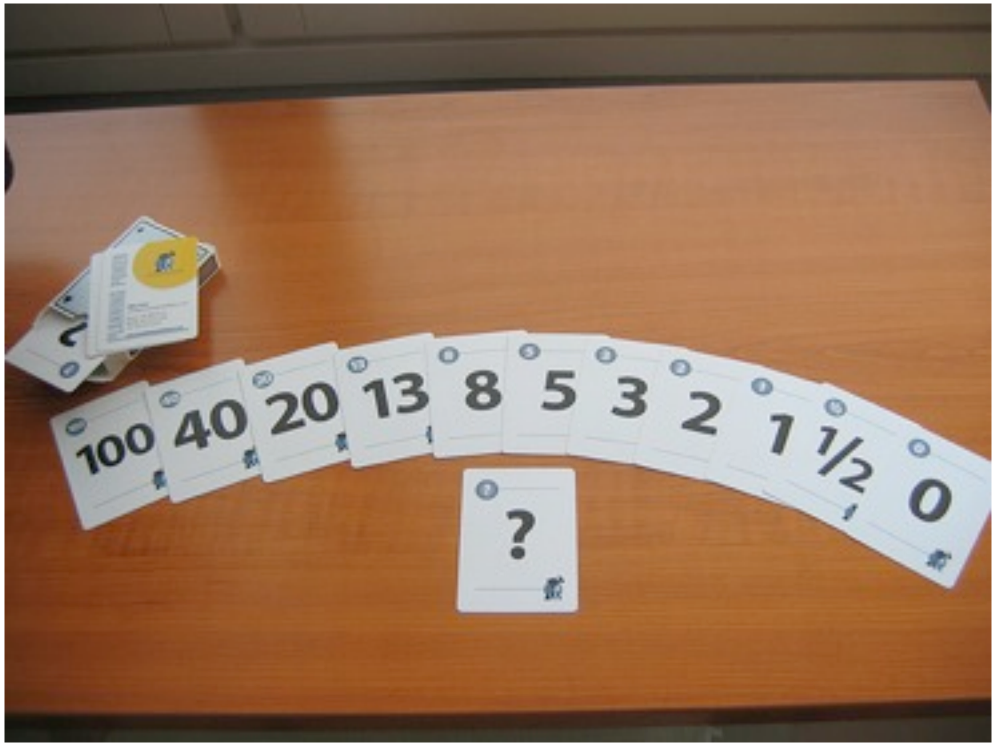
➡ except multiple POs and multiple projects

Scrum Ceremonies: Standup

- Key daily coordination activity for the team.
- Reflects what has been accomplished, what is planned for the day, and identifies impediments.

Scrum Ceremonies: Story Estimation Meetings

- Developers and POs.
- Planning poker helps guide conversation.
- Level of detail increases as work gets closer.



Scrum Ceremonies: Sprint Planning

- Team commits to a set of stories from top of prioritized backlog.
- Consider past performance - "yesterday's weather."

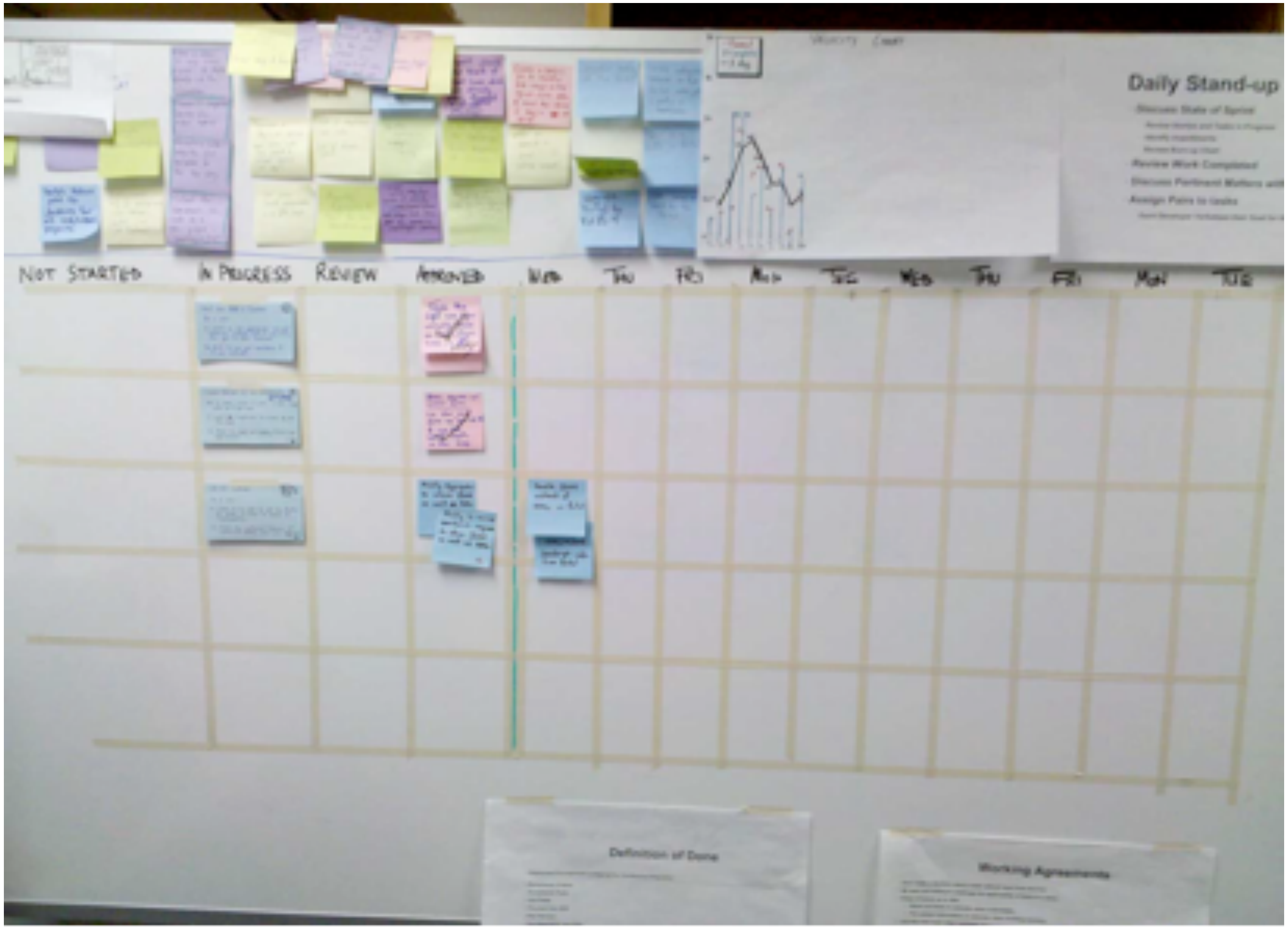
Scrum Ceremonies: Sprint Review

Show it off!

Scrum Ceremonies: Retrospectives

- Inspect and adapt - closes the loop.
- Acknowledge successes.

Scrum Artifacts: Physical Information Radiator



Continuous Integration

- Small increments in functionality, automatically tested.
- Applied to new code, but legacy code yet to be incorporated into CI environment.
- Coordination between teams, especially with regard to database changes.

(Acceptance) Test-Driven Development

Writing tests (executable specifications) for the software *before* the software is written.

(Acceptance) Test-Driven Development

Results in:

- Efficient, complete solutions.
- Solid, readable code.

(Acceptance) Test-Driven Development

...but we learned that it's hard to:

- write good tests,
- learn,
- keep doing TDD.

Pair Programming

- Adjustments for different personalities and programming styles.
- Code benefits.

Mixed pairing/independent code development

- If it's not pair-programmed, it requires code review.
- Focus on streamlining code review process:
  - Smaller checkins.
  - Scrum board signals status.
  - Announcement on IM chat room.

Challenges

Unstable Velocity

Challenges Encountered

- Inconsistent estimation.
- Impacts of holidays, meetings, illness.

Impact

- Lowers PO's confidence.

Unstable Velocity

Resolutions

- Re-evaluating story estimates.
- Comparing new story estimates to established baseline.
- "Bucket" stories at the end of planning meetings.
- Attention to definition of "done" during planning.

Multiple projects in a single team

Challenges Encountered

- How to distribute projects within or across sprints.
- Different numbers of projects assigned to each team (two vs. four).
- Difference in project loads forced different style of management by POs.

Multiple projects in a single team

Resolutions

- Focused on the fewest number of projects possible in a sprint (but this means a project might be put aside for weeks or months while other projects are being worked).
- Developers reorganized from two teams to three.

Adoption of new practices

- e.g. Scrum, Pair programming, TDD.
- Requires consensus from both developers and POs.
- Requires flexibility.

Testing (TDD), Quality and Time

Challenges

- Convincing POs whether long-term benefits are worth the short-term costs.
- Learning TDD.

Testing (TDD), Quality and Time

Resolution

- Find someone to champion the process.
- Negotiate six month trial period.
- Success!

Successes

Customer Satisfaction

(In this case, PO satisfaction.)

Customer Satisfaction

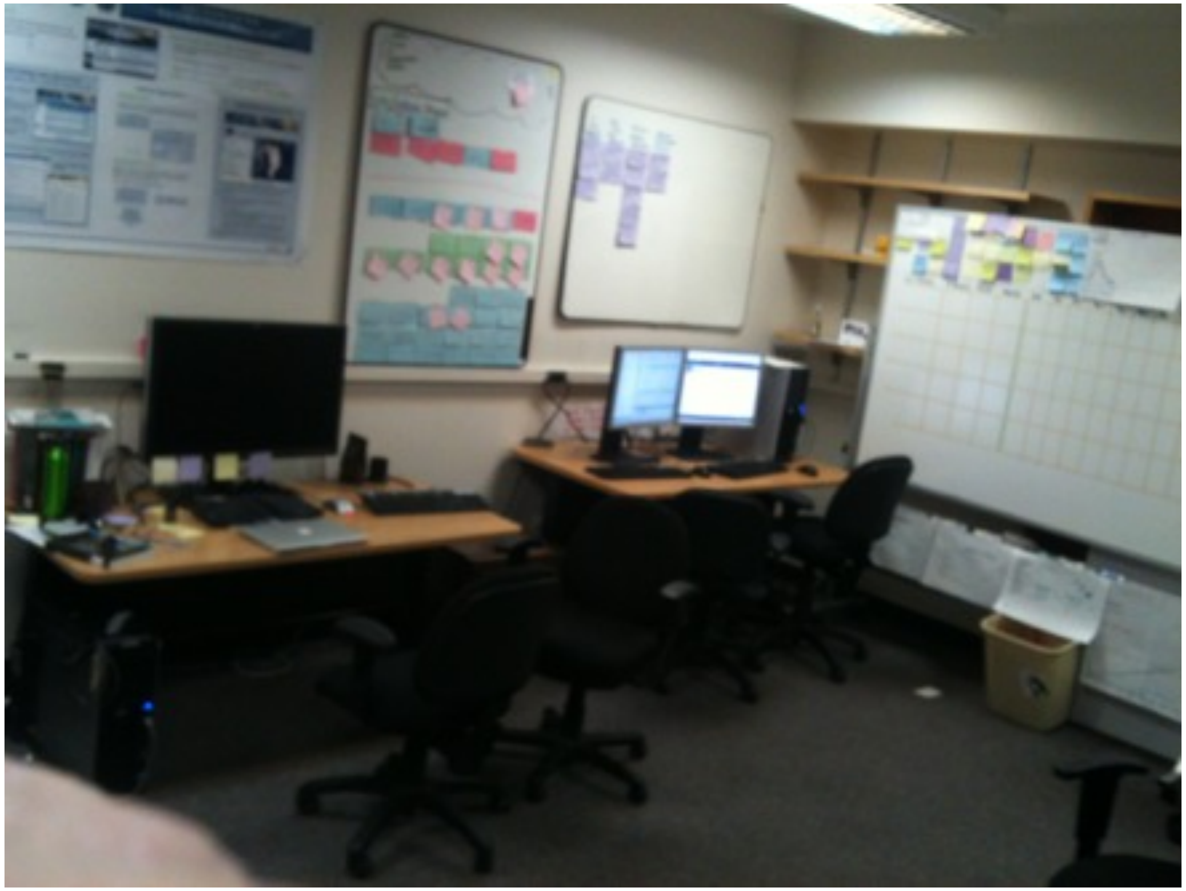
- External deadlines met.
- Successful demos (internal and external).
- Velocity used to effectively gauge future progress.

Highly effective collaboration

- Up to eight people working in same codebase.
- Very little confusion about what people were working on and how it impacted the work others were doing.
- There were times that we stepped on each other, but those issues were recognized and dealt with quickly.
- Never had significant amount of divergent or duplicated work.

Highly effective collaboration

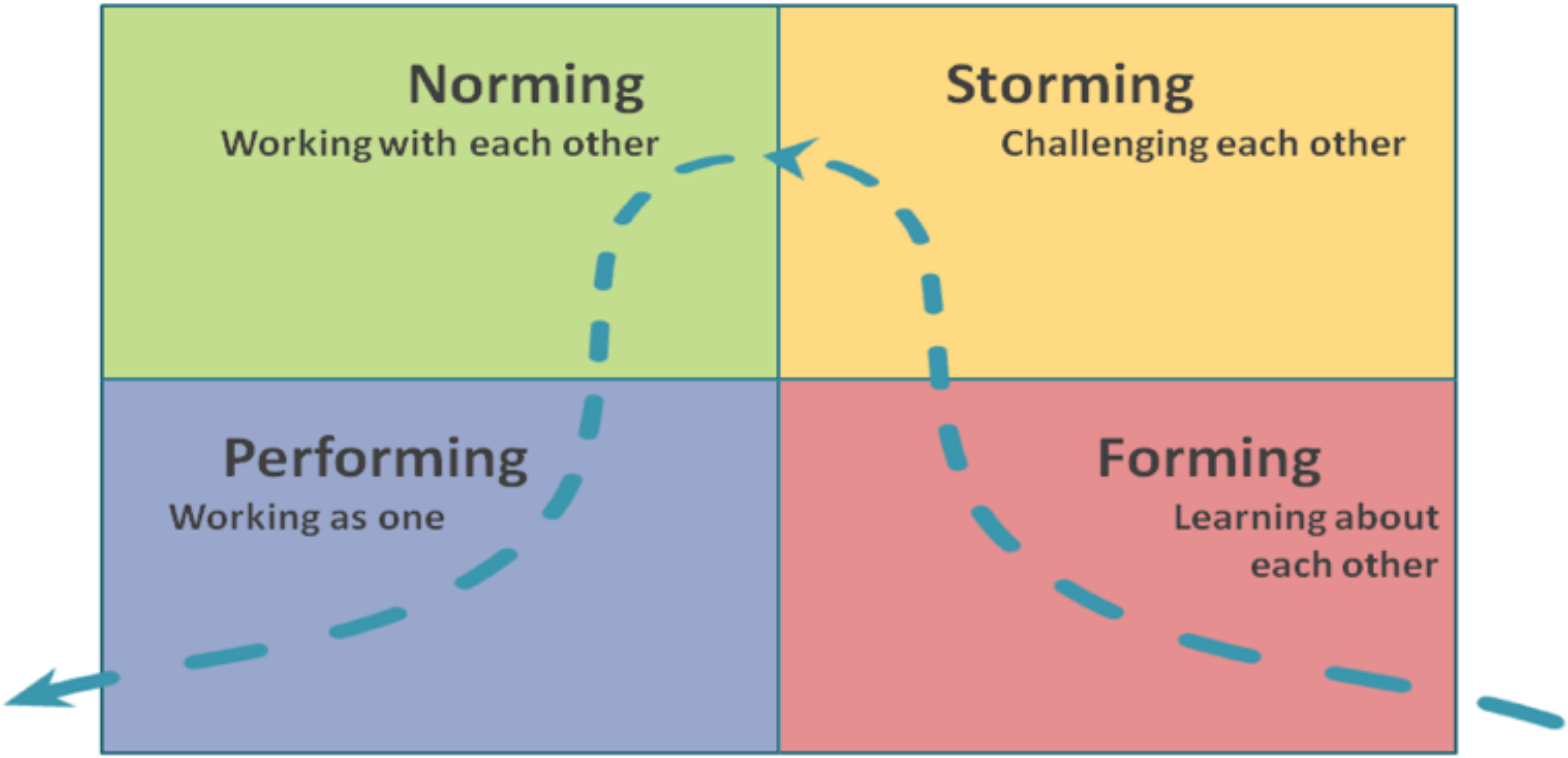
- Co-location and open team space.
- More high-bandwidth conversations, fewer emails.



What Next?

New team structure

Tuckman's model:



Continuous Delivery / Deployment

- Need to standardize lessons from Continuous Integration across teams (even if we're not Continuously Deploying).
- Dependency management for libraries, databases, web services is a hard problem.

Multiple teams, one codebase

Will multiple teams be as successful as a single cohesive team in managing this challenge?

Inter-team Learning

Implement a “Scrum of Scrums” as a mechanism for self-inspection and adaptation.

Agile Science Programming

Goals:

- Knowledge transfer.
- Automation.

Agile Science Programming

Strategy:

- Automated testing.
- Continuous integration.

Agile Science Programming

Intersection of experiences

Science programming, IDL and NumPY vs Agile, web application development and Java.

Agile Science Programming

Other challenges:

- Exploratory approach to data processing makes it difficult to write concise, estimatable stories.
- External deadlines for data products forced some Agile practices to be compromised.

Agile Science Programming

Bottom line:

We're still learning how to deliver "science code" via a Scrum framework.

Contact Info

Email us:

```
[$firstName, ".", $lastName, "@nsidc.org"].concatenate()
```

- **Julia Collins**
- Danielle Harper
- Margaret McNulty
- Jess Lacy
- Luis Lopez
- **Scott Lewis**
- Miao Liu
- **Joseph Oldenburg**
- Stuart Reed
- Donna Scott
- **Ian Truslove**
- Hannah Wilcox

Questions ?

Slides available at http://bit.ly/y6RuoK
or
https://docs.google.com/present/edit?id=0AZQaguDOzK9rZGQ3bWZ4YjlfMjNnZm5nbjhkbQ