prime medicine_

Delivering on the promise of Prime Editing

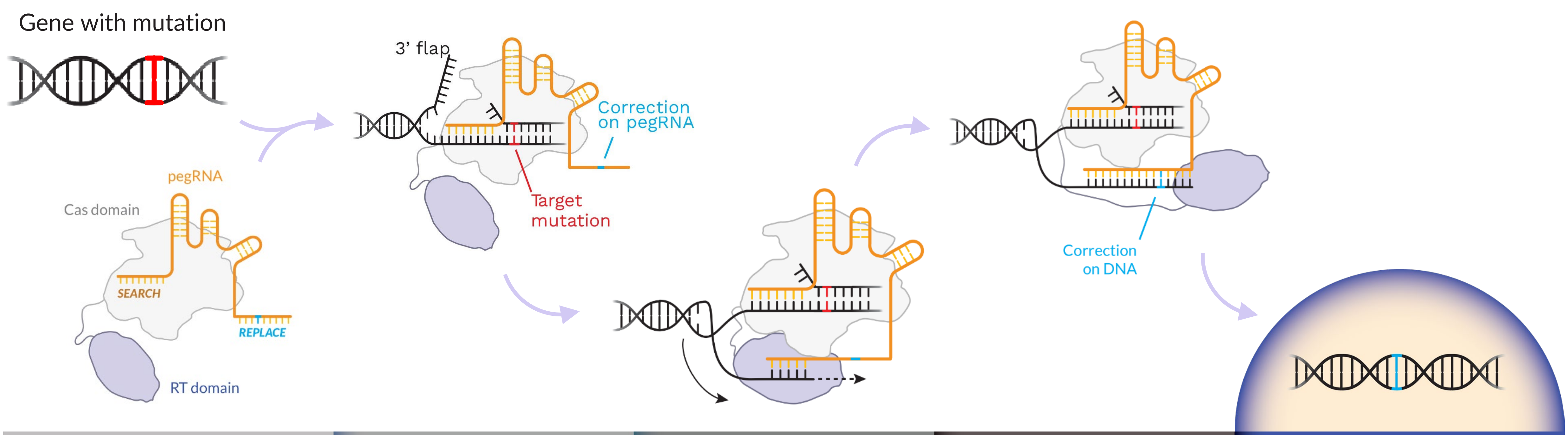
# Integrate to Innovate: Strategic Approaches to Modular Automation

*Future Labs, Automation, & Technology West*
*November 12, 2024*

Rachael Cohen
Associate Director
Prime Medicine

# Prime Editing: a gene editing technology that is programmable for both *search* and *replace*

prime_
medicine




**SEARCH**

Prime editor complex initiates search for target DNA

**FIND & NICK**

Prime editor complex finds DNA with target mutation, nicks one strand

**PRIME**

Nicked DNA strand primes the RT domain for DNA synthesis

**REPLACE**

Prime editor complex copies in corrective DNA sequence

**GENE CORRECTED**

3' flap preferentially incorporated[1], excess flap repaired, gene fully corrected

[1] Completion of an edit requires 3 'edit checks'

pegRNA = prime editing guide RNA; RT = reverse transcriptase; Cas = CRISPR associated protein

Anzalone, et al (David R. Liu). Search-and-replace genome editing without double-strand breaks or donor DNA. *Nature*, 2019.

# Our automation strategy

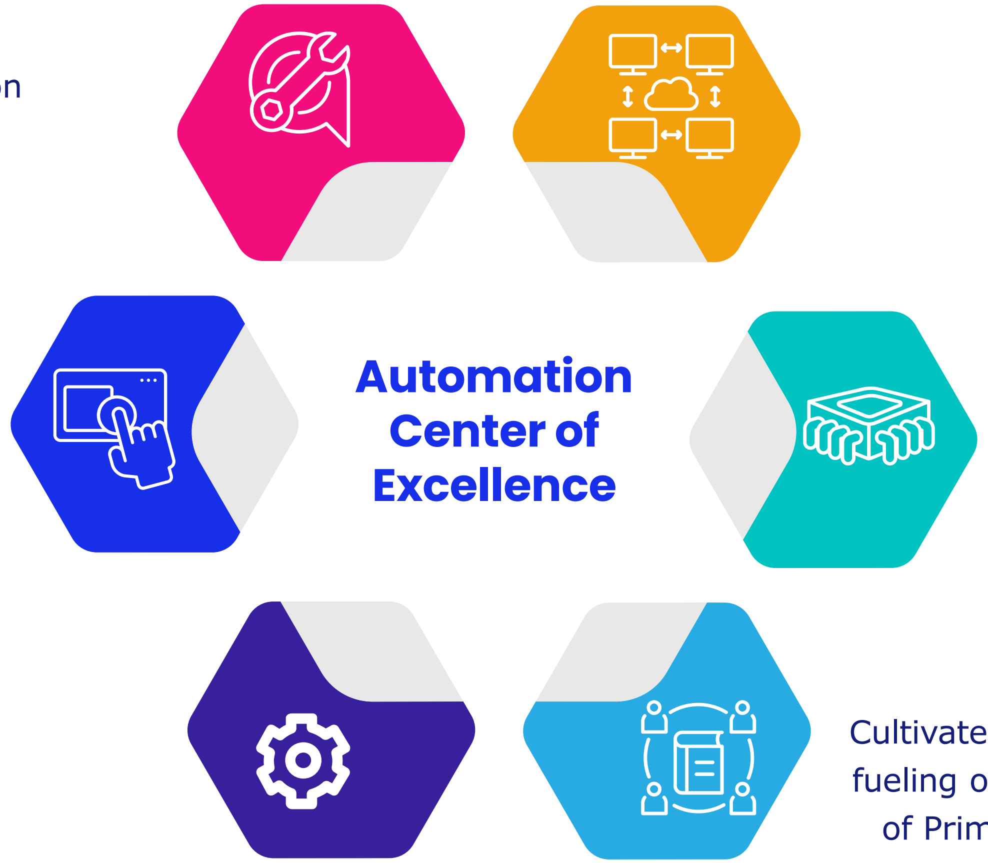
Automated processes to increase our capacity, reduce error and variability, generate and structure our data, and feed into a data pipeline to power innovation

### HARDWARE

Implementing lab instrumentation or systems to labor-intensive or monotonous tasks

### INFORMATICS

Build digital solutions to transform our raw data into information and knowledge




### PROGRAMMING

Adopting a no-code user interface to interact with hardware

### AI/ML

Exploit AI/ML models to learn, mine, and correlate our data to create scientific insights and power innovation

### OPERATIONS

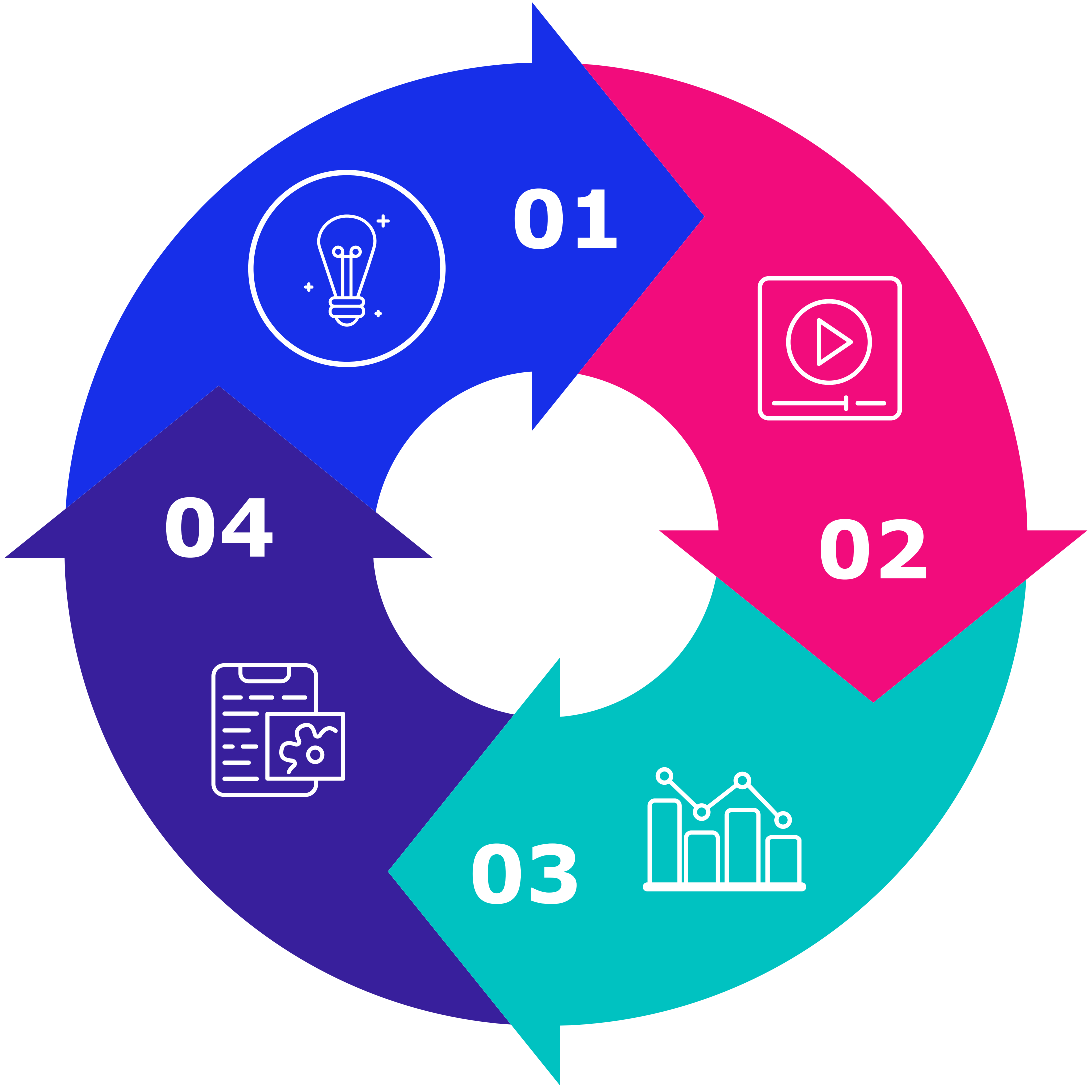
Building out a team to evaluate, develop, deploy, and maintain systems

### CULTURE

Cultivate an automation culture at Prime, fueling our potential to bring the promise of Prime Editing to as many patients as fast as possible.

# Our data pipeline

## A closer look at our past, present, and future




| | Where we started | Where we are | Where we're going |
|---|---|---|---|
| **Design** | • Scientist defined | • Defined by scientist and programmed in "no code" UI | • Generated by AI/ML from previous run |
| **Execution** | • Lack of record<br>• Lack of standardization<br>• 100% manual | • Barcoded consumables<br>• Metadata collection | • Benchling entry triggers automated protocol |
| **Analysis** | • On the fly analysis<br>• Lack of plate metadata retention | • Standardized analysis | • Data file generation launches automated analysis |
| **Results** | • Multi-platform results<br>• Inconsistent structure<br>• Variable link to experiment | • Results manually uploaded to Benchling | • Automated upload of results<br>• Results fuel next experiment through AI/ML |

# Supporting Prime's current and future automation requires well-defined team responsibilities

prime_ medicine

## We've got 10+ integrated systems, including GMP equipment, with more to come

| Build | Install | Support |
| --- | --- | --- |
| Workflow Assessment | Standardization | Incident Response |
| Instrument Evaluation | Liquid Class Development | Metrics Analysis |
| Design | Programming | Programmatic Changes |
| Installation | Teaching | End user training |
| UI Development | Troubleshooting | Maintenance |
| UAT | Training | Repair |

# Agenda

prime_ medicine

## Showcasing how automation and digital infrastructure accelerates bringing Prime Editing to patients

### Build

- **Workflow Assessment**
- Instrument Evaluation
- **Design**
- **Installation**
- UI Development
- UAT

### Install

- **Standardization**
- **Liquid Class Development**
- Programming
- Teaching
- Troubleshooting
- Training

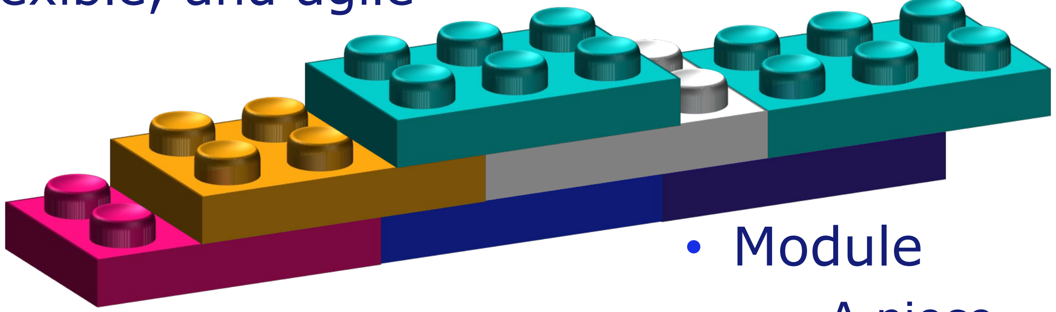
### Support

- Incident Response
- **Metrics Analysis**
- **Programmatic Changes**
- End user training
- Maintenance
- Repair

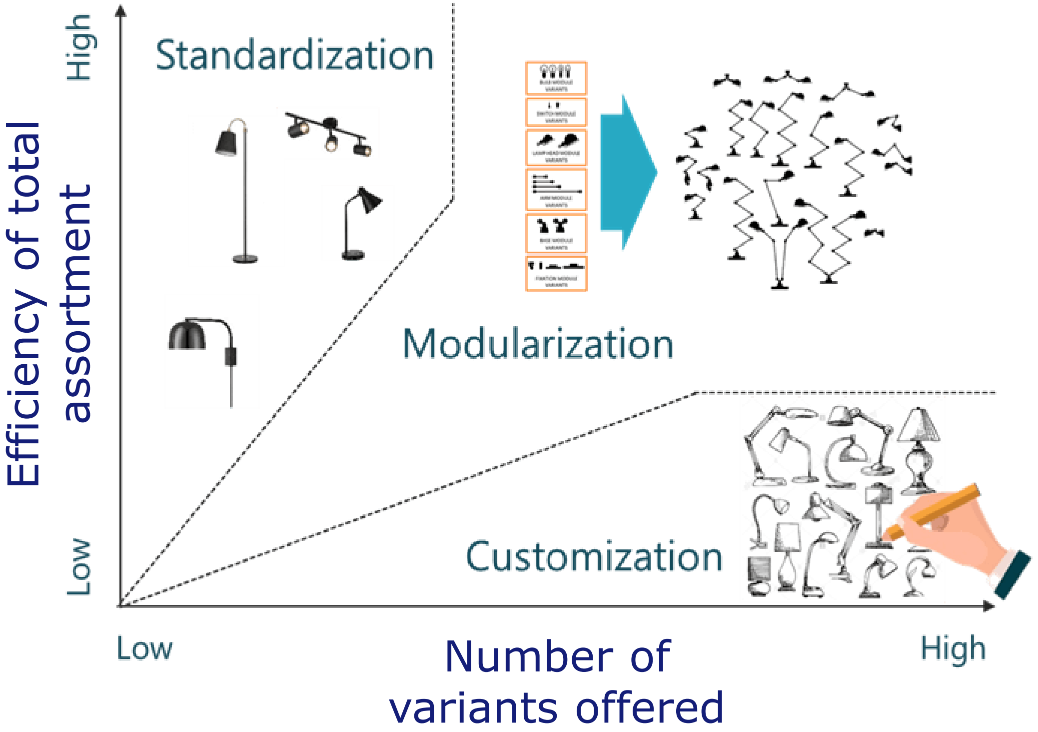
# Module, Modular, Modularity … What do they mean?

## Definitions vary across teams, companies, and fields – so let's define

- Modularity
  - The ability of a system to be efficient, flexible, and agile

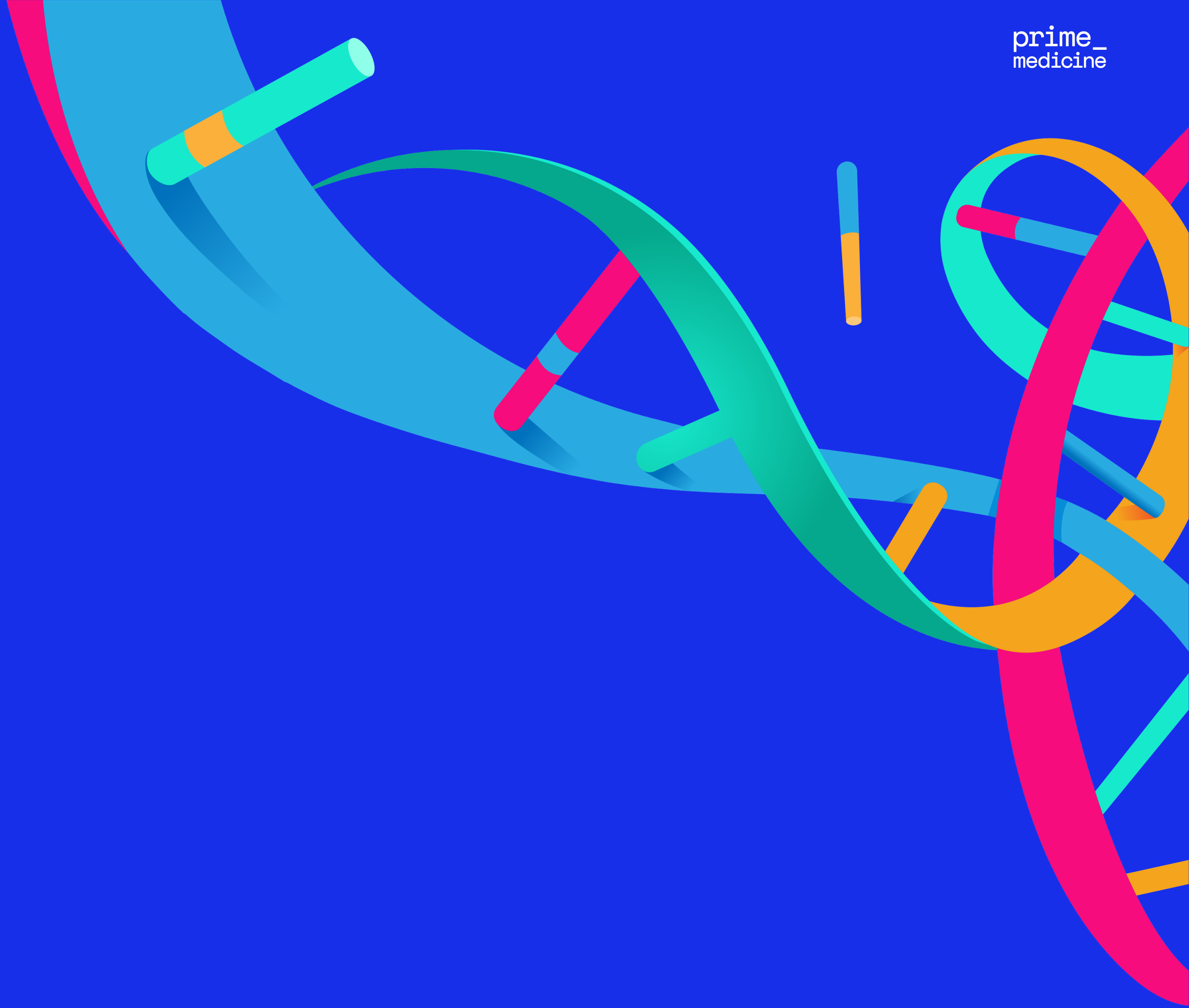


- Module
  - A piece, unit, or part
- Modular
  - A group of modules




1. https://www.modularmanagement.com/blog/all-you-need-to-know-about-modularization

prime_
medicine



# Build

# We design our systems to be modular

prime_ medicine

## But modular can have multiple definitions

### Agility

- Equipment goes down?
  - We've got a backup for that
- Swap this for that?
  - You've got it

### Efficiency

- Standardized features across systems
  - Teaching
  - Liquid classes
  - Programming

### Flexibility

- DIY build and integration
  - We're building tables
  - 3D printing
  - In-house programming

# Modularity begins at design and we’ve streamlined our request system to enable it

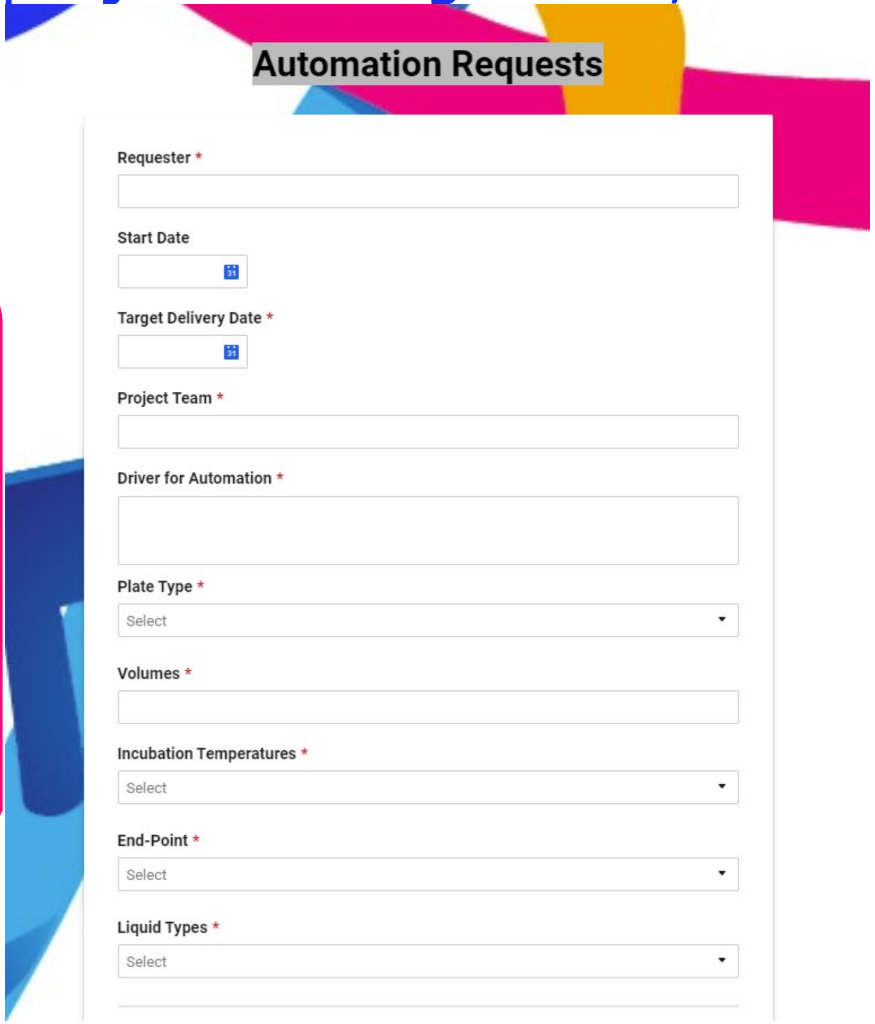


## Utilizing Smartsheet, we’ve developed an all-in-one system for requests, design, project management, and improvements

01

**Request**

Submitted by the end user and alerts the automation team for evaluation



Automation Requests

Requester *

Start Date

Target Delivery Date *

Project Team *

Driver for Automation *

Plate Type *
Select

Volumes *

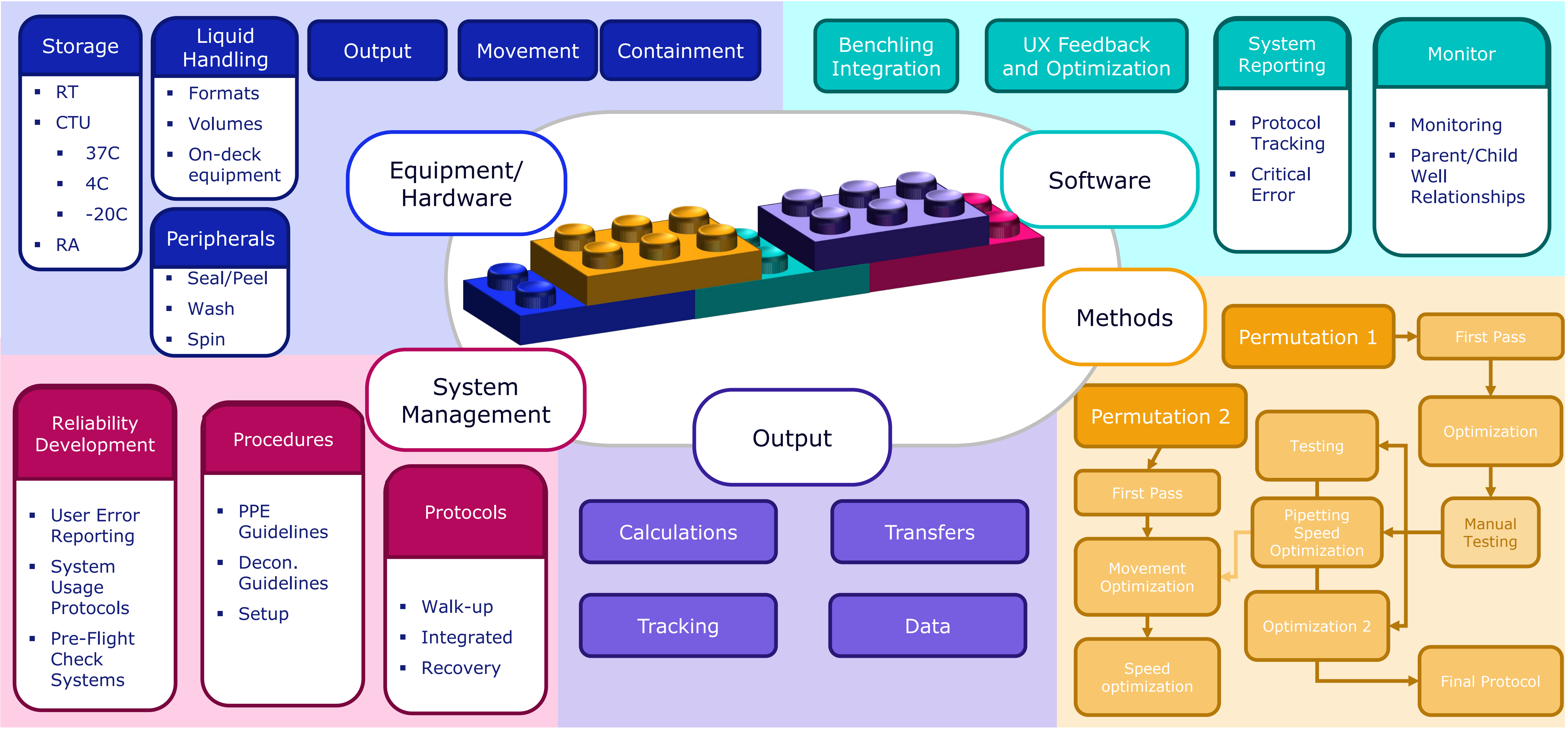
Incubation Temperatures *
Select

End-Point *
Select

Liquid Types *
Select

# Workcell Development – It's more than just labware and equipment

prime_ medicine



Build

Install

Support

# Modularity begins at design and we've streamlined our request system to enable it

prime_
medicine

## Utilizing Smartsheet, we've developed an all-in-one system for requests, design, project management, and improvements

### 01 Request

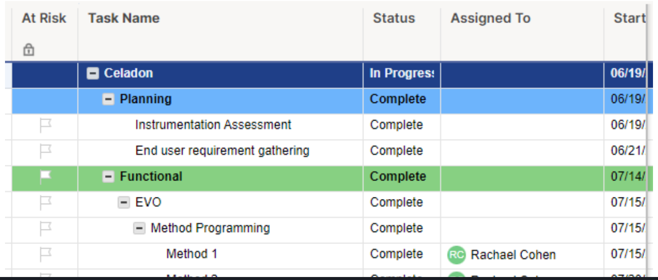
Submitted by the end user and alerts the automation team for evaluation

### 02 Evaluate

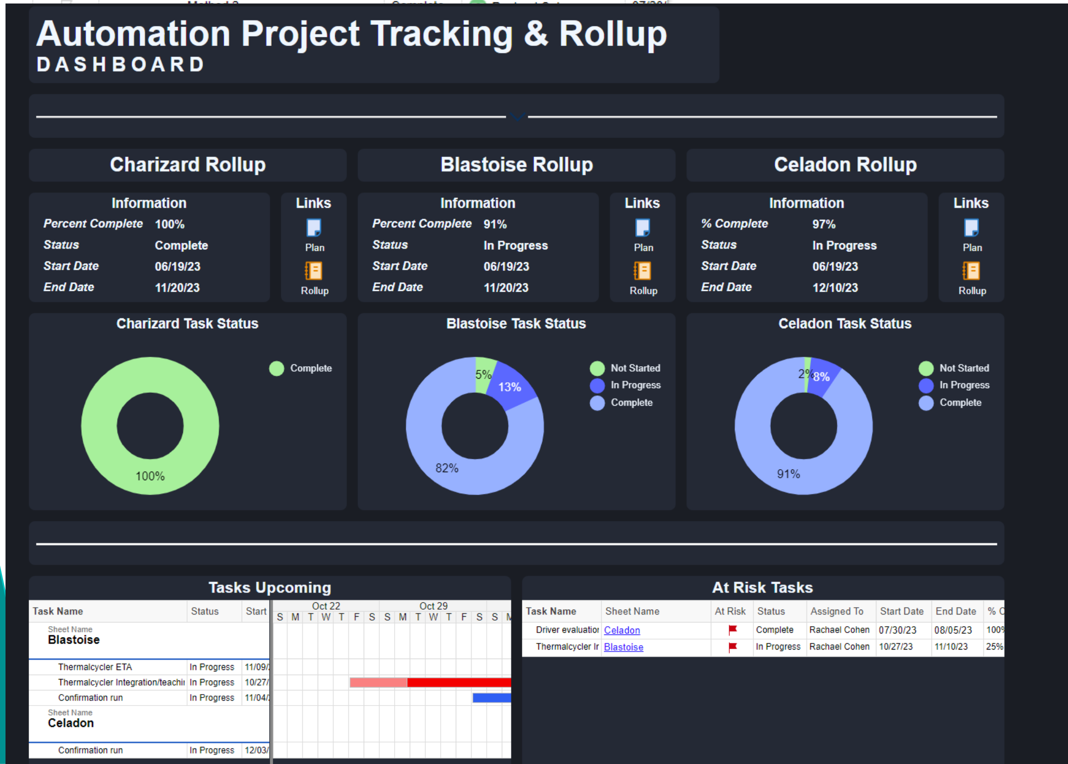
Walkthrough requirements to set requirements, expectations, and milestones

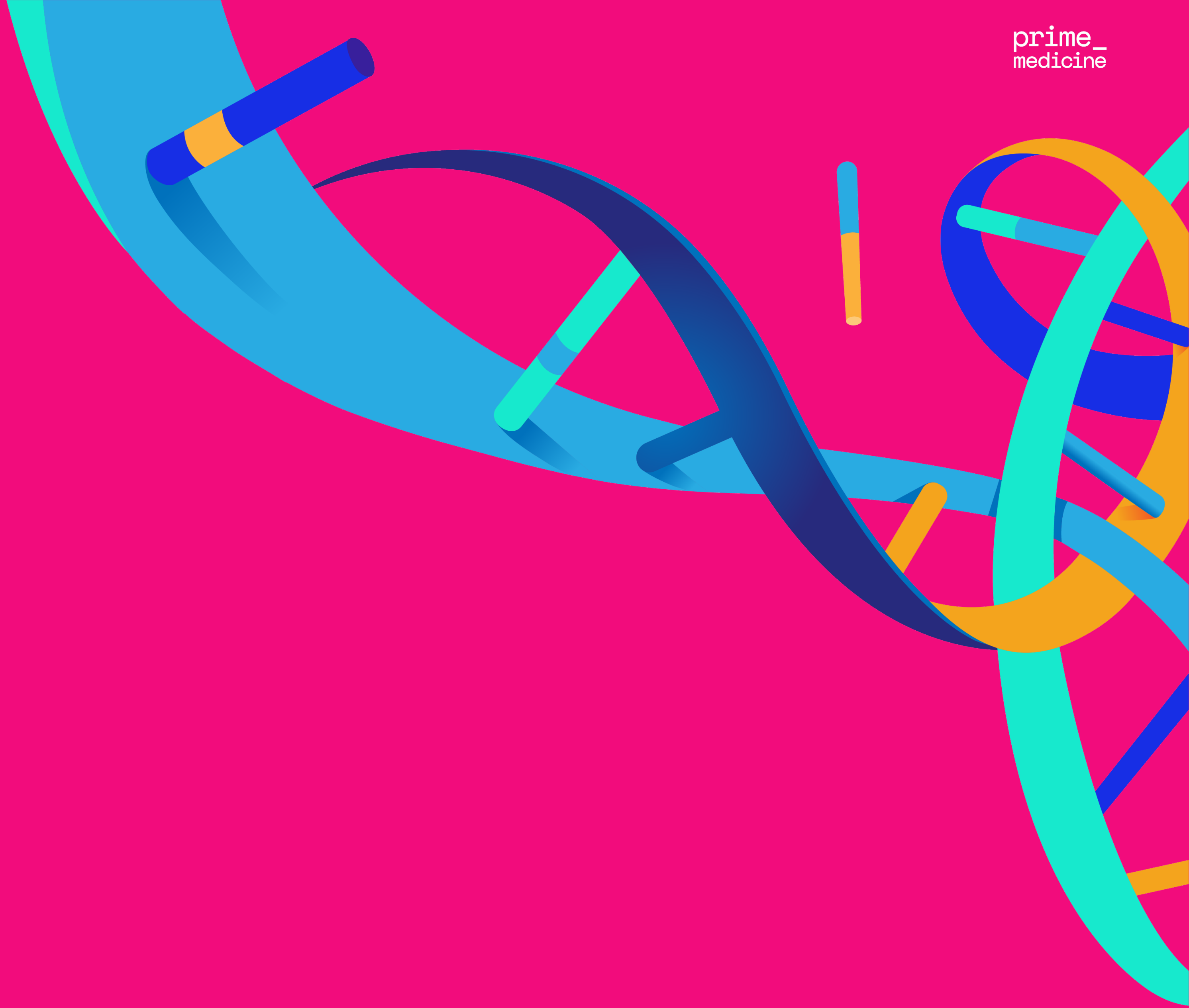
### 03 Plan

Build out the project plan to keep track of deliverables and keep all stakeholders aware



| At Risk | Task Name | Status | Assigned To | Start |
|---|---|---|---|---|
| | Celadon | In Progress | | 06/19/ |
| | Planning | Complete | | 06/19/ |
| | Instrumentation Assessment | Complete | | 06/19/ |
| | End user requirement gathering | Complete | | 06/21/ |
| | Functional | Complete | | 07/14/ |
| | EVO | Complete | | 07/15/ |
| | Method Programming | Complete | | 07/15/ |
| | Method 1 | Complete | Rachael Cohen | 07/15/ |





# Install

Build

Install

Support

# Standardization of procedures enables our ability to adapt

prime_
medicine

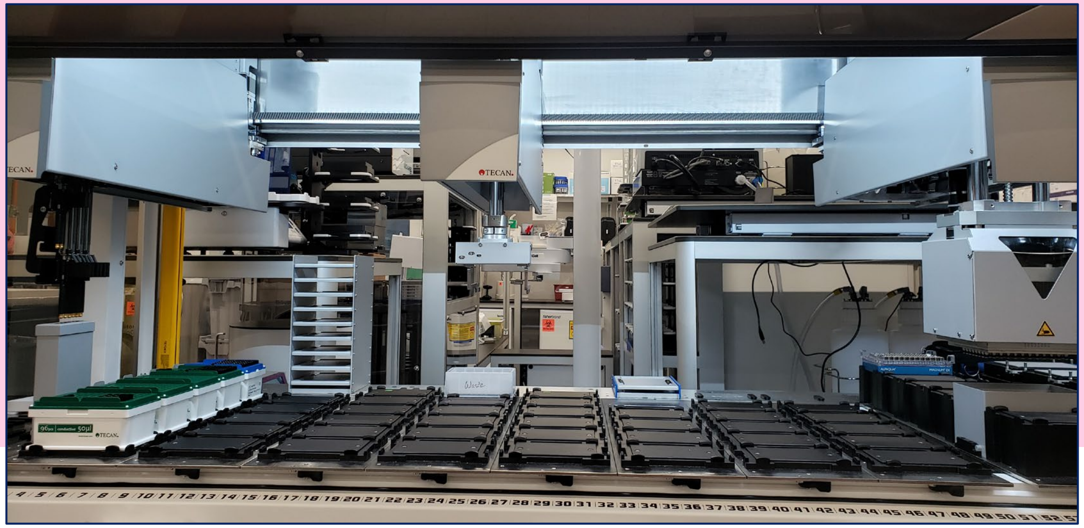
## Harmonization across engineers and equipment is required for smooth transitions

**Labware**

Teaching

Liquid Classes

- We maintain a database of approved labware including critical details
  - Naming convention
  - Manufacturer
  - Catalog Number
    - Problematic lots
  - Dimensions
- New requests are submitted to the team, reviewed, and deployed
  - Newly deployed labware is taught across *all* systems
  - Database is updated



Build
Install
Support

# Standardization of procedures enables our ability to adapt

prime_
medicine

## Harmonization across engineers and equipment is required for smooth transitions

Labware
**Teaching**
Liquid Classes

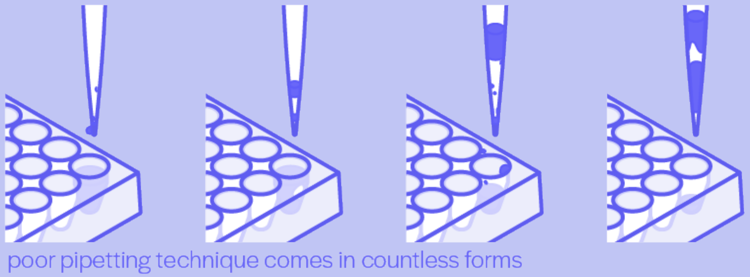
- Liquid handler procedures
  - Labware definitions
  - Vectors
- Robotic Arms
  - Gold standard labware
  - Gold standard positions
  - Path definition

# Standardization of procedures enables our ability to adapt

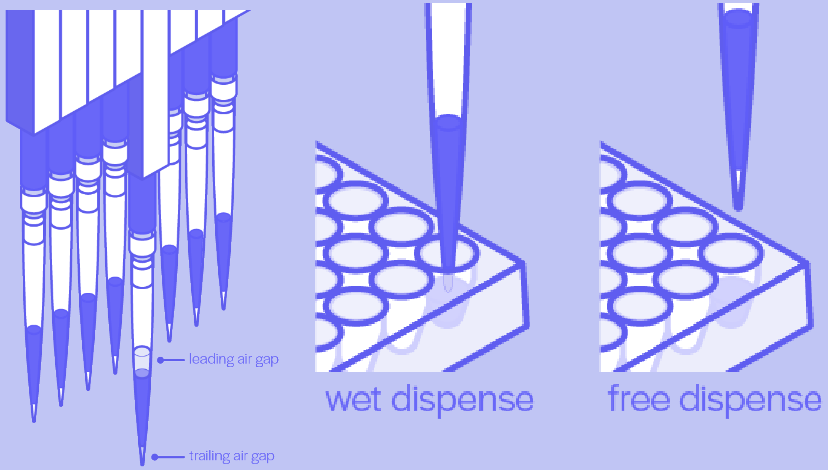
prime_ medicine

## Harmonization across engineers and equipment is required for smooth transitions

Labware
Teaching
**Liquid Classes**




- Standardized liquid classes across systems
  - Naming convention
  - Sub-classes (as appropriate)
  - Purpose
  - Behavior
- Liquid classes are optimized on each type of liquid handler
  - Each class must meet predefined specifications for accuracy and precision
  - Parameters for the class are maintained in a database
- Calibration for each liquid handler is verified at regular intervals



Build
Install
Support

# Liquid class development and optimization is essential for transferability

prime_
medicine

## Artel MVS allows us to not only develop our classes, but track their performance over time

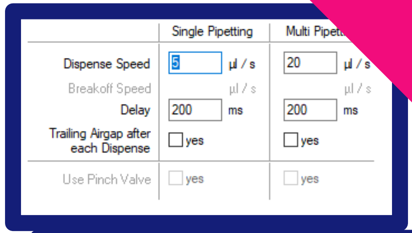
### DEFINE
What liquids?
Properties?
Requirements?

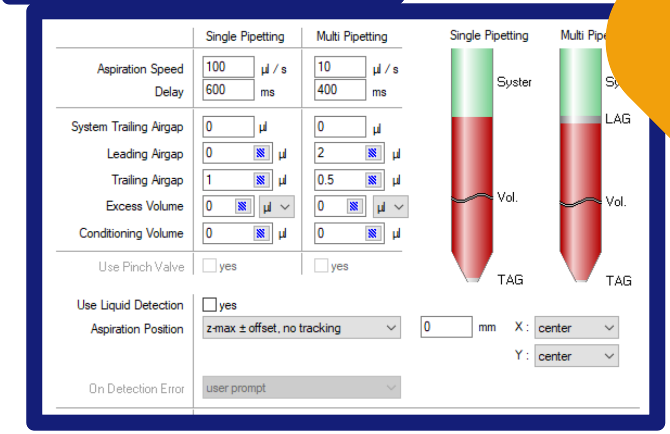
### Measure
Artel MVS
96: 0.1 – 350 uL
384: 0.01 – 55 uL

### CONTROL
Monitor performance regularly

### Design
DOE for effect screening and optimization

### Analyze
Utilize JMP for selection of factors and optimal conditions




# Liquid class development and optimization is essential for transferability

prime_
medicine

## Artel MVS allows us to not only develop our classes, but track their performance over time

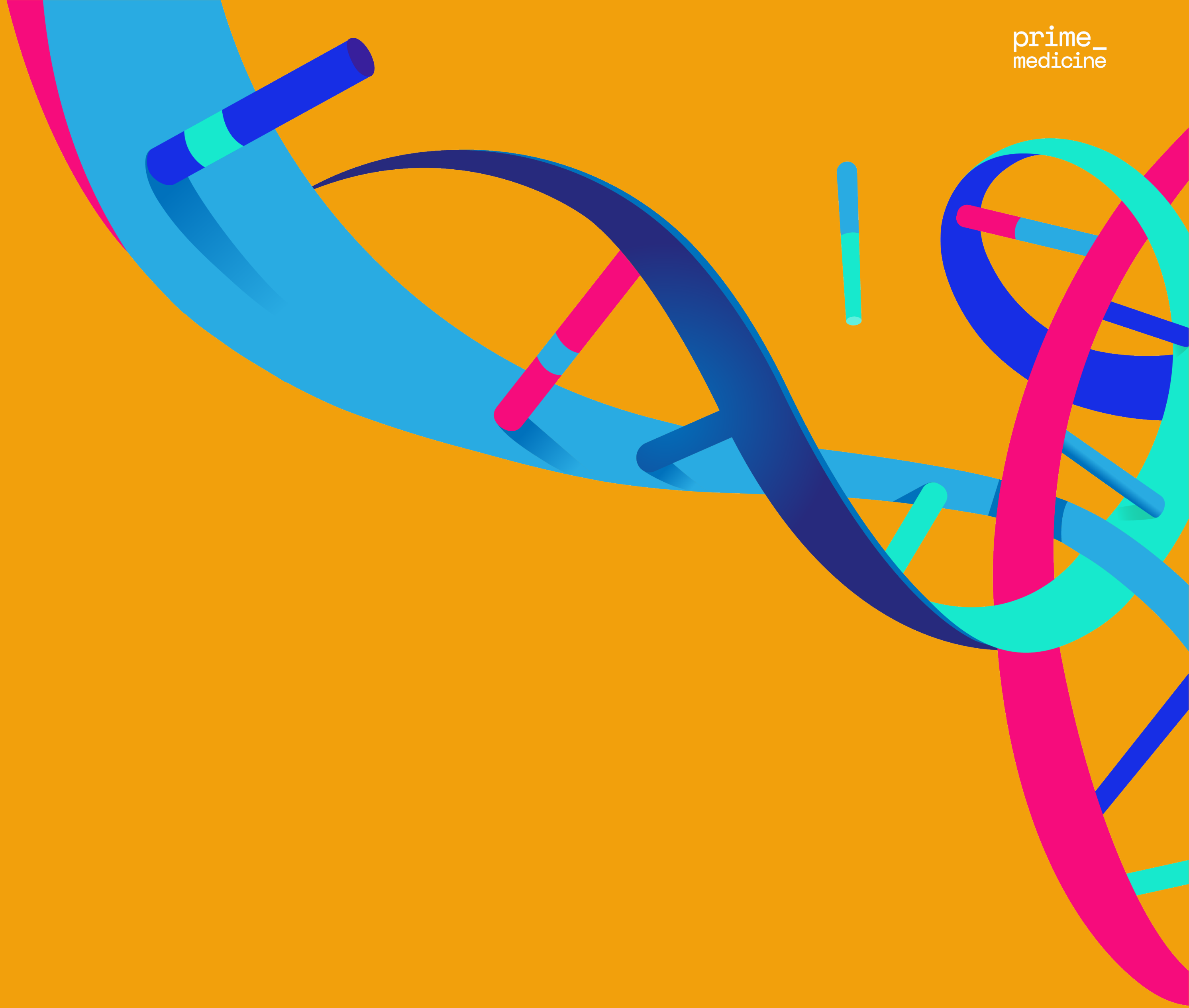
### Bulk Reagents

- Standardized to 3 liquid classes
- Screening of 16 factors
- Optimization of 5 critical factors
- <1% CV and <2% error

| | 6 uL | 13 uL | 20 uL | 27 uL | 34 uL | 41 uL | 48 uL |
|---|---|---|---|---|---|---|---|
| %CV | 0.51% | 0.44% | 0.31% | 0.28% | 0.26% | 0.22% | 0.25% |
| %Error | 1.58% | 1.02% | -0.07% | -0.78% | -0.89% | 0.94% | 1.02% |

### Serial Dilution

- Multi-step dilution causes cumulative error
  - Impacting assay results
- Reduced to 1-2 (liquid handler dependent) classes



# Support

Build

Install

Support

# We collect metrics wherever we can to understand behavior and utilization

prime_ medicine

## Run time data gives us a continuous look at system utilization

1

### Run Time

- Method Start/Stop times
- Number of plates/samples
- Recoveries
- Associated notebooks

- Systems generate run logs during every run, with metadata
  - Scheduler records start and stop of global run
  - Integrated equipment records start and stops of subroutines
- Logs are standardized across all systems
- Utilization metrics fuel reconfiguration and capital requests

# We collect metrics wherever we can to understand behavior and utilization



## Submission data shows us a per sample breakdown of need

- Submissions are required for use of the systems
  - Currently via Benchling or Smartsheet
  - If it's not in a submission, it's not tested
- Submission metrics allow for future projection modeling

**Submissions**

2

- Program
- Submitting team
- Method
- Samples

Build

Install

Support

# We collect metrics wherever we can to understand behavior and utilization

prime_ medicine

## Incident reporting shows us how the systems are behaving

- Incidents occur on systems all the time
  - We collect data every time that happens
  - QR codes in the lab allow analysts to report issues at time of issue
  - Engineer response requires an incident
- We can track hardware, software, facility, and more root causes

**Incidents**

- System
- Equipment
- Error
- Sample Loss

3

# We collect metrics wherever we can to understand behavior and utilization

## Firmware data gives us a snapshot of mileage



4

**Firmware**

- Movements
- Errors
- Volumes
- Tip pickups

- Monthly, firmware is collected from all available instruments
  - Data collected is dependent on the instrument
- Firmware allows for prediction of failure before failure occurs

# We collect metrics wherever we can to understand behavior and utilization



## The more data the better

# Summary

# We're modular for the lifecycle of a system

## From design to maintenance, we build and support automation to innovate for our future

How early planning and preparation sets modularity up for success

How modularity can impact system design and build

How standardization and optimization enhances performance

How metrics collection enables modularity by allowing us to understand factors

Delivering on the promise of Prime Editing

# Thank you!

primemedicine.com