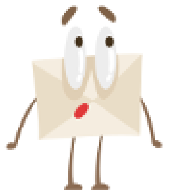
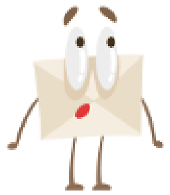
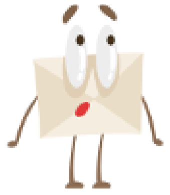
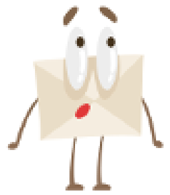
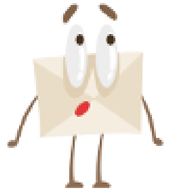
# Handling Failure in RabbitMQ

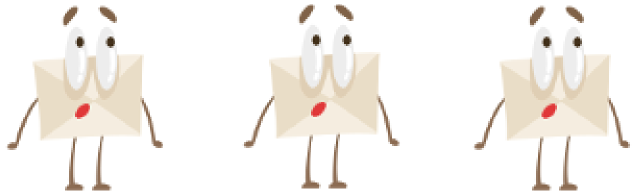
Lorna Mitchell, IBM
https://speakerdeck.com/lornajane

# Queues and RabbitMQ

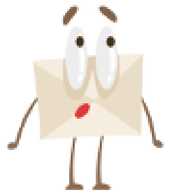
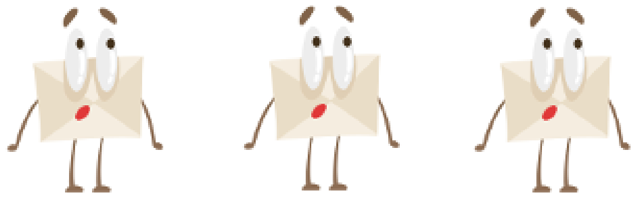
- Queues are a brilliant addition to any application
- They introduce coupling points
- RabbitMQ is an open source, powerful message queue
- https://www.rabbitmq.com





## What is Failure?

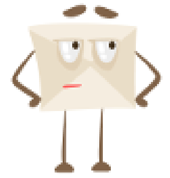
Reality.

## A Selection Box Of Failures

# Message Not Processed

**Question:** Better late than never?



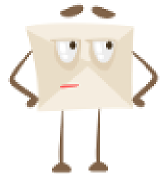
@lornajane

## Message Not Processed

**Question:** Better late than never?

If not:

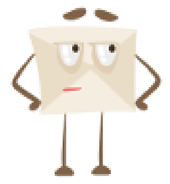
- set up "at-most-once" delivery
- configure queue with `auto-ack`



## Message Not Processed

To react to unprocessed messages:

- set up "at-least-once" delivery; requires messages to be acknowledged
- beware duplicate and out-of-order messages
- if the consumer drops connection or dies, message will be requeued automatically
- detect failure and reject messages with `requeue`, or implement retries

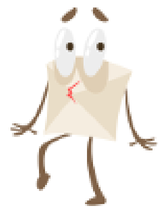


@lornajane

# Implementing Retries

If there isn't built-in support, try this:

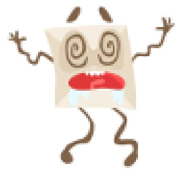
1. Identify message should be retried
2. Create a **new** message with same data
3. Add retry count/date
4. Ack the original message
5. Reject after X attempts



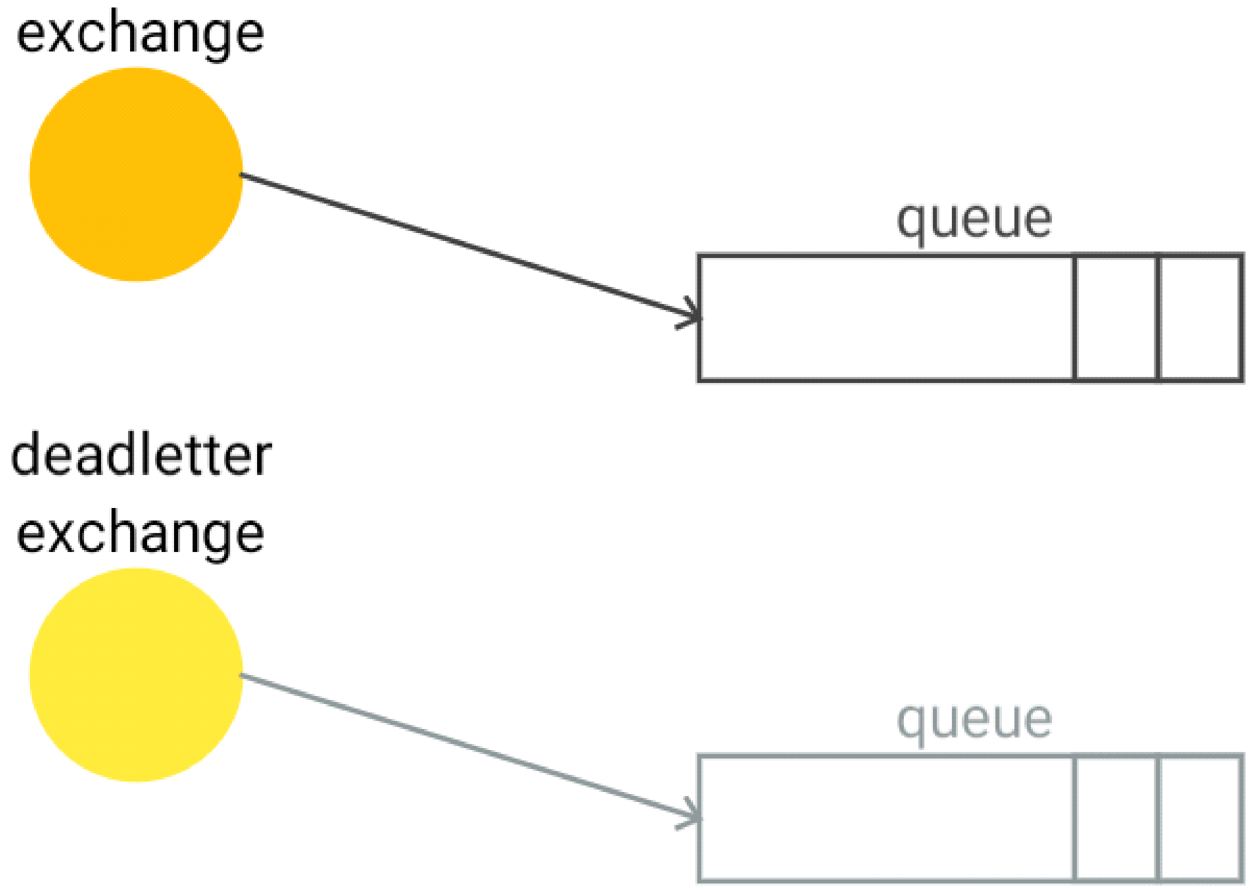
# Can Never Process Message

When a worker cannot process a message:

- be defensive and if in doubt: exit
- reject the message (either with or without `requeue`)
- look out for "poison" messages that can never be processed
- configure the queue with a "dead letter" exchange to catch rejected messages



# Dead Letter Exchanges



@lornajane

# Reincarnating Messages

From the dead letter exchange we usually:

- monitor and log what arrives
- collect messages, then re-route to original destination when danger has passed

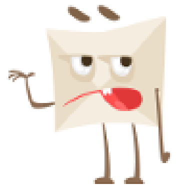
@lornajane

## Queue Is Getting Bigger

A constantly-growing queue should set off alarms

Ideal queue length depends on:

- size of message
- available consuming resources
- how long a message spends queued



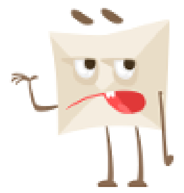
@lornajane

## Queue Is Getting Bigger

To stop queues from growing out of control:

- set max queue size (*oldest messages get dropped when it gets too long*)
- set TTL on the message to let stale messages get out of the backlog

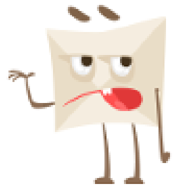
In both cases, we can use the dead letter exchange to collect and report on these



@lornajane

## Many Queues, Many Workers

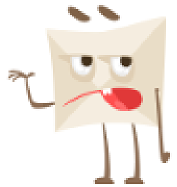
- Deploy as many workers as you need, they may consume multiple queues
- The "right" number of workers may change over time
- Workers can be multi-skilled, handling multiple types of message
- If in doubt: use **more** queues in your setup



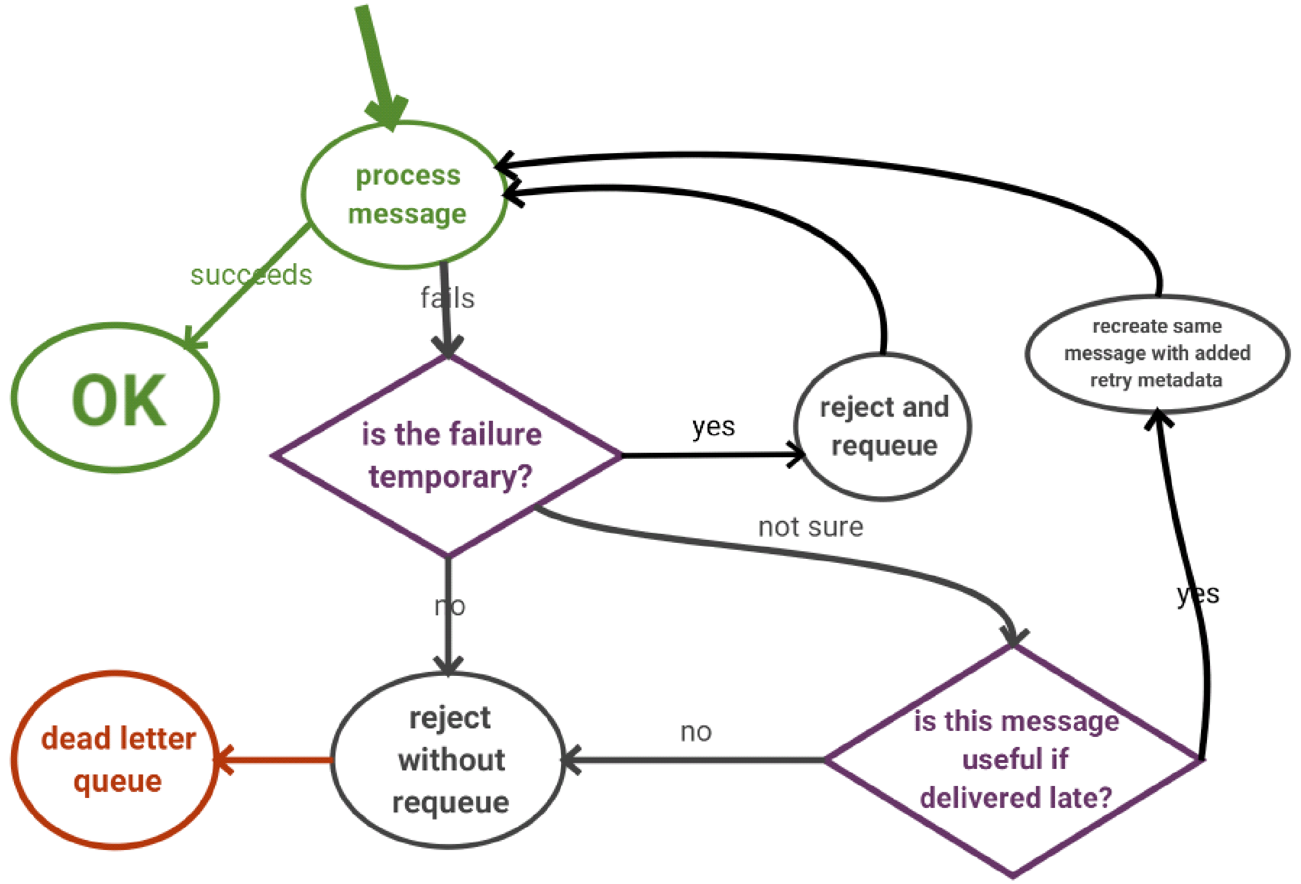
# Healthy Queues

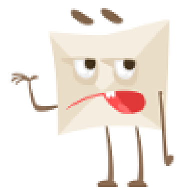
Good metrics avoid nasty surprises

As a minimum: queue size, worker uptime, processing time



# Choose How To Fail


process message
succeeds
OK
fails
is the failure temporary?
yes
reject and requeue
not sure
no
reject without requeue
dead letter queue
is this message useful if delivered late?
no
yes
recreate same message with added retry metadata



@lornajane

# Thanks!

Blog post: http://lrnja.net/rabbitfail

Personal blog: https://lornajane.net

Try RabbitMQ:

- https://rabbitmq.com/
- https://ibm.cloud/