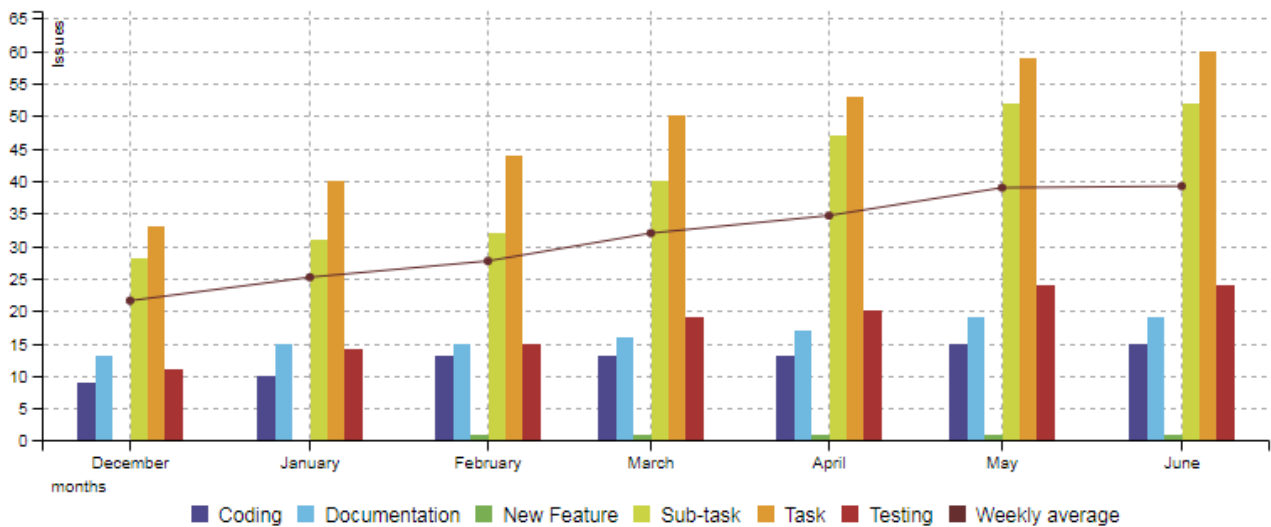


Open issues with average

This chart displays all issues selected by a specific JQL and a time period. The results get displayed as a bar chart. In addition, the average for each period get's displayed as a line chart.

[approve](#) [Download Scripted Chart Bundle](#)

Chart preview



Parameters

Name	Type	Default
Jql_Param	JQL autocomplete	
Period_Param	Time Period Picker	Month

Layout Script

JavaScript Template

```
function formatWeek(d){
  if (d instanceof Date)
  {
    // Copy date so original doesn't get modified
    d = new Date(Date.UTC(d.getFullYear(), d.getMonth(), d.getDate()));

    // Set to nearest Thursday: current date + 4 - current day number
    // Make Sunday's day number 7
    d.setUTCDate(d.getUTCDate() + 4 - (d.getUTCDay()||7));

    // Get first day of year
    var yearStart = new Date(Date.UTC(d.getUTCFullYear(),0,1));

    // Calculate full weeks to nearest Thursday
    var weekNo = Math.ceil(( (d - yearStart) / 86400000) + 1)/7);

    // Return array of year and week number
```

```

        return [d.getUTCFullYear(), weekNo];
    }
    return '';
}

function formatTooltipAsHours(value, ratio, id, index)
{
    return value.toFixed(2) + ' h';
}

function formatQuarter(d)
{
    if (d instanceof Date)
    {
        var q = d.getMonth();
        q = parseInt(q / 3) + 1;
        return 'Q' + q;
    }
    return '';
}

function formatHalfyear(d)
{
    if (d instanceof Date)
    {
        var q = d.getMonth();
        q = parseInt(q / 6) + 1;
        return 'H' + q;
    }
    return '';
}

chartData.types = { "Weekly average" : "line" };

var c3arg = {
    data: chartData,
    grid: {
        y: {
            show: true
        },
        x: {
            show: true
        }
    },
    axis: {
        x: {
            type: 'timeseries',
            label: {
                text: chartData.custom.xLabel,
                position: 'outer-left'
            },
            tick: {
                format: eval(chartData.custom.xTickFormat),
                culling: {
                    max: 25
                },
                fit: true,
                multiline: false
            }
        },
        y: {
            label: chartData.ytype
        }
    }
};

if (chartData.custom && chartData.custom.tooltip)
{
    var tooltipFunction = eval(chartData.custom.tooltip);
    c3arg.tooltip = {

```

```

        format: {
            value: tooltipFunction
        }
    };
}

if ( chartData.custom && chartData.custom.gauge_max)
{
    c3arg.gauge = {
        max: parseFloat(chartData.custom.gauge_max)
    };
}

c3.generate(c3arg);

```

Data Script

Groovy Script

```

import java.lang.reflect.Field;
import java.math.BigDecimal;
import java.util.Arrays;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;

import org.apache.log4j.Logger;
import org.apache.lucene.document.Document;

import com.atlassian.jira.component.ComponentAccessor;
import com.atlassian.jira.issue.DocumentIssueImpl;
import com.atlassian.jira.issue.Issue;
import com.atlassian.jira.jql.parser.JqlParseException;
import com.atlassian.jira.jql.parser.JqlQueryParser;
import com.atlassian.query.Query;
import com.decadis.jira.xchart.api.ChartParam;
import com.decadis.jira.xchart.api.model.Period;
import com.decadis.jira.xchart.api.util.DateUtils;
import com.decadis.jira.xchart.grouping.GroupValueExtractor;
import com.decadis.jira.xchart.grouping.GroupValueExtractorFactory;
import com.decadis.jira.xchart.model.ChartData;
import com.decadis.jira.xchart.model.MetaCountGroup;
import com.decadis.jira.xchart.templates.TemplateConstants;
import com.decadis.jira.xchart.utils.ChartUtilsImpl;
import com.decadis.jira.xchart.utils.DateUtilsImpl;
import java.io.IOException;
import java.util.Collections;
import com.atlassian.adapter.jackson.ObjectMapper;
import com.atlassian.jira.user.ApplicationUser;
import com.atlassian.jira.util.I18nHelper;
import com.atlassian.jira.util.IOUtil;
import com.decadis.jira.xchart.api.Chart;
import com.decadis.jira.xchart.api.util.ChartBuilder;
import com.decadis.jira.xchart.api.util.FilterUtils;
import com.decadis.jira.xchart.impl.ChartBuilderImpl;
import com.decadis.jira.xchart.utils.FilterUtilsImpl;

Logger logger = Logger.getLogger("Unresolved incl. average - xChart");
JqlQueryParser jqlQueryParser = ComponentAccessor.getComponent(JqlQueryParser.class);
Query query = jqlQueryParser.parseQuery(Jql_Param);
I18nHelper i18n = ComponentAccessor.getJiraAuthenticationContext().getI18nHelper();
logger.debug("JQL=" + Jql_Param);

try

```

```

{
    query;
} catch (JqlParseException e)
{
    logger.warn("Bad JQL:" + jql, e);
    throw new IllegalArgumentException("Bad JQL: " + jql);
}

Period selectedPeriod = Period.fromString(Period_Param);
logger.debug("Period=" + selectedPeriod);

MetaCountGroup countGroup = new MetaCountGroup();
DateUtils dateUtils = new DateUtilsImpl(ComponentAccessor.getJiraAuthenticationContext().getLocale());
GroupValueExtractor groupValueExtractor = GroupValueExtractorFactory.Create("issuetype");
Field documentField;
HashMap map = new HashMap();

documentField = DocumentIssueImpl.class.getDeclaredField("document");
documentField.setAccessible(true);

for ( Issue issue : new FilterUtilsImpl().performSearch(query, user) )
{
    for ( String group : groupValueExtractor.getGroups((Document) documentField.get(issue)) )
    {
        String periodName = dateUtils.getPeriodGroup(issue.getCreated(), selectedPeriod);
        if(map.get(periodName)) {
            BigDecimal a = map.get(periodName);
            a = a.add(BigDecimal.ONE);
            map.put(periodName,a);
        } else {
            map.put(periodName,BigDecimal.ONE);
        }
        group = groupValueExtractor.getResolvedValue(group, issue);
        countGroup.addValue(BigDecimal.ONE, group, periodName);
    }
}

for(Entry e : map.entrySet()){
    countGroup.addValue(getAverage(e.getKey(),e.getValue()),"Weekly average",e.getKey())
}
//countGroup.fillMissingValues();
countGroup.accumulateGroups();

ChartData chartData = new ChartData(i18n.getText("common.concepts.issues"));
chartData.setDateFormat(DateUtils.SimpleDateFormatD3);
chartData.setPeriod(selectedPeriod);
chartData.setType("bar");

ChartUtilsImpl.TransformResult(countGroup, chartData);
return chartData;

BigDecimal getAverage(String period, BigDecimal value){
    double daysInMonth = getDaysInMonth(DateUtils.SimpleDateFormat.parse(period));
    double weeks = daysInMonth / 7;
    double average = value.doubleValue() / weeks;
    return BigDecimal.valueOf(average).setScale(2,1);
}

int getDaysInMonth(Date d){
    // Create a calendar object and set year and month
    Calendar mycal = new GregorianCalendar(d.getYear(),d.getMonth(),d.getDay());

    // Get the number of days in that month
    int daysInMonth = mycal.getActualMaximum(Calendar.DAY_OF_MONTH);
    return (double)daysInMonth;
}

```



Related examples

Title
Using Jira Software specific classes and Pickers in Scripts
Story status category grouped by Epic and custom value
Simple Timeseries Chart
Simple Table Report
Simple Scripting Example
Report - Lucene Group By
Open issues with average
Issues in specific status (Period)
Group ordering in scripted charts
Gantt Diagram
Customers in a Google Map
Created vs. resolved with trend
Comments count by user in JQL result
Block Search
2Y Axes Chart
Simple External Database Chart

If you still have questions, feel free to refer to our [support](#) team.